



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

KONTROL KESEIMBANGAN UPPER BODY PADA PROSES AKSELERASI DAN DESELERASI GERAKAN BERJALAN ROBOT SEPAK BOLA HUMANOID

Anas Mufid Nurrochman
NRP 2213100056

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Muhtadin ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - TE 141599

**KONTROL KESEIMBANGAN *UPPER BODY* PADA
PROSES AKSELERASI DAN DESELERASI GERAKAN
BERJALAN ROBOT SEPAK BOLA *HUMANOID***

Anas Mufid Nurrochman
NRP 2213100056

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Muhtadin ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017

Halaman ini sengaja dikosongkan



FINAL PROJECT - TE141599

**UPPER BODY BALANCING CONTROL OF HUMANOID
ROBOT SOCCER WALKING MOTION ON THE PROCESS OF
ACCELERATION AND DECELERATION**

Anas Mufid Nurrochman
NRP 2213100094

Advisor
Dr. Ir. Djoko Purwanto, M.Eng.
Muhtadin S.T., M.T.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Kontrol Keseimbangan *Upper Body* pada Proses Akselerasi dan Deselerasi Gerakan Berjalan Robot Sepak Bola Humanoid”**

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2017

Anas Mufid Nurrochman
NRP.2213 100 056

Halaman ini sengaja dikosongkan

**KONTROL KESEIMBANGAN UPPER BODY PADA
PROSES AKSELERASI DAN DESELERASI GERAKAN
BERJALAN ROBOT SEPAK BOLA HUMANOID**

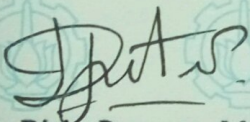
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember**

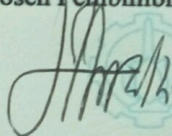
Menyetujui:

Dosen Pembimbing I



Dr. Ir. Djoko Purwanto, M.Eng.
NIP. 196512111990021002

Dosen Pembimbing II



Muhtadin, S.T., M.T.
NIP. 198106092009121003



**SURABAYA
JULI, 2017**

Halaman ini sengaja dikosongkan

ABSTRAK

Nama : Anas Mufid Nurrochman
Judul : Kontrol Keseimbangan *Upper Body* pada
Proses Akselerasi dan Deselerasi Gerakan
Berjalan Robot Sepak Bola Humanoid
Pembimbing : 1. Dr. Ir. Djoko Purwanto, M.Eng
2. Muhtadin ST., MT.

Penelitian pada Tugas Akhir ini bertujuan untuk mengembangkan Robot Sepak Bola *Humanoid*. Dalam penelitian ini dilakukan pembuatan robot *humanoid* dengan derajat kebebasan 20 *DoF* dan tinggi 88 cm. Salah satu hal yang paling penting dalam pertandingan robot sepak bola adalah kecepatan robot untuk mengejar bola. Pada saat melakukan akselerasi atau deselerasi kecepatan berjalan, bagian badan robot akan cenderung mempertahankan posisi awal sehingga keseimbangan robot akan terpengaruh. Dalam penelitian ini dilakukan penambahan proses akselerasi dan deselerasi pada robot dan penambahan kontrol keseimbangan pada pinggul (*hip*) robot untuk menjaga keseimbangan badan robot saat melakukan pergerakan akselerasi atau deselerasi. Berdasarkan hasil penelitian didapatkan desain mekanik robot yang dapat berjalan lurus dengan kecepatan 0,23 m/s, berjalan kesamping dengan kecepatan 0,75 m/s, dan berputar dengan kecepatan 30 derajat setiap langkahnya. Penerapan kontrol PD dengan metode *metode trial and eror* dengan nilai P sebesar 0,04 dan D sebesar 0,02 pada aktuator robot bagian *hip pitch* saat berjalan dengan proses akselerasi dapat membuat robot lebih stabil saat berjalan dengan mencapai steady state dalam 2.5 detik

Kata Kunci : robot *humanoid*, akselerasi, deselerasi, kontrol keseimbangan

Halaman ini sengaja dikosongkan.

ABSTRACT

Name : Anas Mufid Nurrochman
Title : *Upper Body Balancing Control of Humanoid Robot Soccer Walking Motion on the Process of Acceleration and Deceleration*
Advisor : 1. Dr. Ir. Djoko Purwanto, M.Eng
2. Muhtadin ST., MT.

The goal of this research is to make humanoid robot with 20 degrees of freedom and height of 88 cm. One of the most important things in a soccer robot match is the speed of the robot to catch the ball. At the time of acceleration or speed deceleration runs, the body of the robot will tend to maintain the initial position so that the robot balance will be affected. In this research, the acceleration and deceleration process are added on the hip joint of robot to control the balance of robot body during the movement of acceleration or deceleration. Based on the results of the research, a mechanical robot design can run straight with speed of 0.23 m/s, runs at a rate of 0.75 m/s, and rotates at 30 degrees per step. Application of PD control using trial and error method with P value of 0.04 and D of 0.02. According on testing result, robots are more stable when walking by reaching steady state in 2.5 second.

Keywords : *Humanoid Robot, acceleration, deceleration, balance control*

Halaman ini sengaja dikosongkan.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan berkah, rahmat, serta hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul: Kontrol Keseimbangan *Upper Body* pada Proses Akselerasi dan Deselerasi Gerakan Berjalan Robot Sepak Bola Humanoid

Penelitian ini diusulkan sebagai lanjutan dari penelitian tentang robot humanoid pemain sepak bola. Selain itu, penelitian ini dilakukan sebagai prasyarat penulis untuk mendapatkan gelar Sarjana Teknik dari Jurusan Teknik Elektro ITS, bidang studi Elektronika. Penulis ingin menyampaikan terimakasih kepada:

1. Keluarga penulis, khususnya bapak dan ibu yang senantiasa memberikan dukungan baik dukungan material maupun non material.
2. Bapak Ardyono Priyadi, S.T., M.Eng., Dr.Eng . selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
3. Bapak Dr. Ir. Djoko Purwanto, M.Eng. dan Bapak Muhtadin, ST., MT. selaku dosen pembimbing yang selalu memberikan saran serta bantuan dalam penelitian ini.
4. Bapak ibu dosen pengajar Jurusan Teknik Elektro, khususnya bidang studi Elektronika.
5. Seluruh teman-teman beserta dosen pembimbing Tim Robotika ITS
6. Seluruh anggota tim ICHIRO yang telah membantu dalam pengerjaan robot.

Kesempurnaan hanya milik Allah SWT, penyusunan tugas akhir ini tentu masih banyak kekurangan. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat terutama untuk tim ICHIRO. Semoga dengan penelitian ini performa dari robot sepakbola ITS meningkat.

Surabaya, Juni 2017

Penulis

Halaman ini sengaja dikosongkan.

DAFTAR ISI

1. Pendahuluan	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Sistematika Penulisan	3
2. Tinjauan Pustaka	5
2.1 Robot <i>Humanoid</i>	5
2.2 <i>Robocup</i> 2016	6
2.3 Kontes Robot Sepak Bola Indonesia (KRSBI) 2017	7
2.4 Robot <i>Humanoid</i> Darwin-OP	8
2.5 Platform Perangkat Lunak Robot <i>Humanoid</i> Darwin-OP [3]	9
2.6 Robot <i>Humanoid</i> Nimbro-OP[6]	11
2.7 Pola Berjalan Robot Humanoid	12
2.8 Kontroler PID	14
2.8.1 Kontroler Proporsional	15
2.8.2 Kontroler Proporsional Derivatif	16
2.8.3 Kontroler Proporsional Integral	16
2.8.4 Kontroler Proporsional Integral Derivatif	17
2.8.5 Metode Tuning PID [5]	18
2.8.5.1 Metode Ziegler-Nichols	18
2.8.5.2 Metode Cohen-Coon	18
2.8.5.3 Metode <i>Trial and Error</i>	19
2.9 Sistem Operasi Ubuntu	19
2.10 Kontroler CM 730	20

3. Perancangan Sistem dan Implementasi Desain	23
3.1 Desain Mekanik Robot	23
3.1.1 Kepala	25
3.1.2 Badan	26
3.1.3 Lengan	27
3.1.4 Kaki	28
3.1.5 Telapak Kaki	28
3.2 Perancangan <i>Hardware</i> Elektronik Sistem	29
3.3 Perancangan Software Sistem Keseimbangan pada Ake- lerasi dan Deselerasi Robot	30
3.3.1 Akelerasi dan Deselerasi Robot	30
3.3.2 Sistem Keseimbangan Robot	32
3.3.3 Kontroler PD pada bagian <i>Hip</i> robot	32
3.3.4 Kontroler PD pada lengan robot	33
3.3.5 Sistem Keseimbangan pada Akelerasi dan Dese- lerasi Robot	34
3.4 Implementasi Desain Mekanik	35
3.4.1 Kondisi Sebelum Perakitan	36
3.4.2 Kondisi Setelah Perakitan	37
3.4.3 Pengukuran CoM	38
4. Pengujian dan Analisa	39
4.1 Pengujian Mekanik	39
4.1.1 Gerakan Berjalan Lurus	40
4.1.2 Gerakan Berputar	41
4.1.3 Gerakan Berjalan ke samping	42
4.2 Pengujian Keseimbangan Berjalan pada Akselerasi dan Deselerasi Robot	43
4.2.1 Data Hasil Pengujian Keseimbangan berjalan Ro- bot tanpa Akselerasi, deselerasi, dan sistem kese- imbangan	44
4.2.2 Data Hasil Pengujian Keseimbangan berjalan Ro- bot dengan Akselerasi dan deselerasi tanpa sistem keseimbangan	45
4.2.3 Data Hasil Pengujian Keseimbangan berjalan Ro- bot dengan Akselerasi dan deselerasi dengan sis- tem keseimbangan <i>hip</i>	46

4.2.4	Data Hasil Pengujian Keseimbangan berjalan Robot dengan Akselerasi, deselerasi, dan sistem keseimbangan tangan	47
4.2.5	Data Hasil Pengujian Keseimbangan berjalan Robot dengan Akselerasi, deselerasi, dan sistem keseimbangan <i>hip</i> dan tangan	48
4.3	Analisa Hasil Pengujian Sistem Keseimbangan Berjalan Robot	49
4.3.1	Perbandingan Kondisi berjalan Robot Saat Mulai Berjalan dari Keadaan Berhenti	50
4.3.2	Perbandingan Kondisi berjalan robot saat diperlambat	53
4.3.3	Perbandingan Kondisi berjalan robot saat dihentikan	57
5.	Penutup	63
5.1	Kesimpulan	63
5.2	Saran	63
	DAFTAR PUSTAKA	65
	LAMPIRAN	67
	BIOGRAFI PENULIS	75

Halaman ini sengaja dikosongkan.

DAFTAR GAMBAR

2.1	Gambaran pertandingan robot sepak bola <i>humanoid</i>	8
2.2	Tampilan Darwin-OP dan Spesifikasi Dimensi (©Robotis)	9
2.3	Bagan Framework Darwin	10
2.4	Tampilan NimBro OP (©Robotis)	12
2.5	Kondisi DDP dan DSP	13
2.6	Diagram blok kontroler secara umum	14
2.7	Diagram blok kontroler P	16
2.8	Diagram blok kontroler PD	17
2.9	Diagram blok kontroler <i>PID</i>	17
2.10	Logo sistem operasi ubuntu	20
2.11	Kontroler CM 740	21
3.1	Distribusi aktuator dan diagram derajat kebebasan robot Ithaaro	26
3.2	Ukuran dan derajat kebebasan kepala robot	27
3.3	Desain dan ukuran badan robot	27
3.4	Desain mekanik dan diagram derajat kebebasan lengan robot	28
3.5	Desain mekanik dan diagram derajat kebebasan kaki robot	29
3.6	Desain mekanik dan telapak kaki robot	29
3.7	Bagan sistem elektronik robot	30
3.8	Flowchart algoritma akselerasi dan deselerasi robot	31
3.9	Diagram blok akselerasi dan deselerasi robot	32
3.10	Diagram blok sistem keseimbangan robot dengan kontrol PD pada bagian <i>hip pitch</i>	32
3.11	Flowchart sistem keseimbangan robot	33
3.12	Flowchart sistem keseimbangan tangan robot	34
3.13	Diagram blok sistem keseimbangan robot dengan akselerasi	35
3.14	Kondisi part robot sebelum dirakit	35

3.15	Kondisi part robot setelah dirakit	37
3.16	proses pengukuran tinggi pusat massa	38
4.1	Proses pengujian gerakan berjalan lurus ke depan	40
4.2	Proses pengujian gerakan berjalan berputar	41
4.3	Proses pengujian gerakan berjalan ke samping	42
4.4	Lintasan Pengujian Robot	43
4.5	pengujian Pengujian Robot	44
4.6	Data sensor <i>accelerometer</i> Robot berjalan tanpa Akse- rasi, deselerasi, dan sistem keseimbangan	45
4.7	Data sensor <i>accelerometer</i> Robot berjalan dengan Akse- lerasi dan deselerasi tanpa sistem keseimbangan	46
4.8	Data sensor <i>accelerometer</i> Robot berjalan tanpa Aksele- rasi dan deselerasi dengan sistem keseimbangan <i>hip</i>	47
4.9	Data sensor <i>accelerometer</i> Robot berjalan dengan Akse- lerasi dan deselerasi dengan sistem keseimbangan tangan .	48
4.10	Data sensor <i>accelerometer</i> Robot berjalan dengan Akse- lerasi dan deselerasi dengan sistem keseimbangan <i>hip</i> dan tangan	49
4.11	Data sensor <i>accelerometer</i> Robot berjalan tanpa Aksele- rasi, deselerasi, dan sistem keseimbangan	50
4.12	Data sensor <i>accelerometer</i> Robot berjalan dengan Akse- lerasi dan deselerasi, tanpa sistem keseimbangan	51
4.13	Data sensor <i>accelerometer</i> Robot berjalan dengan Akse- lerasi dan deselerasi, dengan kontrol keseimbangan <i>hip</i> . .	51
4.14	Data sensor <i>accelerometer</i> Robot berjalan dengan Akse- lerasi dan deselerasi, dengan kontrol keseimbangan tangan	52
4.15	Data sensor <i>accelerometer</i> Robot berjalan dengan Aksele- rasi dan deselerasi, dengan kontrol keseimbangan <i>hip</i> dan tangan	52
4.16	Perbandingan data sensor <i>accelerometer</i> pengujian Robot berjalan saat akselerasi	53
4.17	Data sensor <i>accelerometer</i> Robot berjalan diperlambat tan- pa Akselerasi, deselerasi, dan kontrol keseimbangan	54
4.18	Data sensor <i>accelerometer</i> Robot berjalan diperlambat de- ngan Akselerasi dan deselerasi, tanpa kontrol keseimbang- an	54

4.19	Data sensor <i>accelerometer</i> Robot berjalan diperlambat dengan Akselerasi dan deselerasi, dan kontrol keseimbangan <i>hip</i>	55
4.20	Data sensor <i>accelerometer</i> Robot berjalan berjalan diperlambat dengan Akselerasi dan deselerasi, dan kontrol keseimbangan tangan	55
4.21	Data sensor <i>accelerometer</i> Robot berjalan tanpa Aksele-rasi, deselerasi, dan kontrol keseimbangan	56
4.22	Perbandingan data sensor <i>accelerometer</i> pengujian Robot saat berjalan dengan deselerasi	56
4.23	Data sensor <i>accelerometer</i> Robot dihentikan tanpa Akse-lerasi, deselerasi, dan kontrol keseimbangan	58
4.24	Data sensor <i>accelerometer</i> Robot dihentikan dengan Ak-selerasi dan deselerasi tanpa kontrol keseimbangan	58
4.25	Data sensor <i>accelerometer</i> Robot dihentikan dengan Ak-selerasi dan deselerasi dengan kontrol keseimbangan <i>hip</i> .	59
4.26	Data sensor <i>accelerometer</i> Robot dihentikan dengan Ak-selerasi dan deselerasi dengan kontrol keseimbangan tangan	59
4.27	Data sensor <i>accelerometer</i> Robot dihentikan dengan Ak-selerasi dan deselerasi dengan kontrol keseimbangan <i>hip</i> dan tangan	60
4.28	Perbandingan data sensor <i>accelerometer</i> pengujian Robot berjalan saat akan berhenti.	60

Halaman ini sengaja dikosongkan.

DAFTAR TABEL

3.1	Tabel nilai dimensi hasil proses desain robot	24
3.2	Spesifikasi aktuator dengan input tegangan 14.8V	25
3.3	Rincian berat part robot sebelum dirakit	36

Halaman ini sengaja dikosongkan.

BAB 1

PENDAHULUAN

1.1 Latar belakang

Kemajuan teknologi robotika berkembang dengan pesat. Hal tersebut dipengaruhi oleh kebutuhan bidang-bidang lain akan robotika yang semakin meningkat, seperti bidang medis, industri, sipil, pemerintahan, manajemen bencana, bahkan olahraga. Penelitian-penelitian di bidang robotika yang disesuaikan dengan bidang-bidang tersebut juga semakin banyak. Demikian juga penelitian tentang robot humanoid. Robot humanoid digunakan untuk kepentingan riset pada berbagai bidang ilmu pengetahuan dengan tujuan bermacam-macam, mulai dari robot untuk kesenian, olahraga hingga robot untuk penyelamatan bencana. Peran robot humanoid yang anatominya dibuat mirip dengan anatomi tubuh manusia di bagian tertentu sangatlah penting untuk menyesuaikan tugasnya.

Dalam perkembangannya, sebagai salah satu tolok ukur perkembangan robot humanoid di dunia, setiap tahun dilakukan pertandingan robot humanoid untuk bermain sepak bola. Pada tahun 1997, dilakukan kompetisi robot sepak bola pertama di dunia dengan nama Robocup. Event tersebut dilakukan untuk menggenjot perkembangan robot humanoid melalui kompetisi. Dengan adanya kompetisi tersebut diharapkan peneliti-peneliti dari berbagai belahan dunia bersaing untuk mengembangkan robot humanoid untuk kepentingan ilmu pengetahuan dan juga diharapkan bisa diaplikasikan ke bidang ilmu lain.

Dalam rangka mengembangkan Robot Sepak Bola humanoid, dalam penelitian ini dilakukan pembuatan robot humanoid dengan 20 DoF dan tinggi 88 cm. Robot didesain meniru anatomi manusia. Anggota gerak pada robot seperti kaki atau anggota gerak bagian bawah lainnya juga didesain menyerupai anggota tubuh manusia. Namun, karena kompleksnya anggota gerak manusia, dalam pengembangannya Robot Sepak Bola humanoid hanya meniru beberapa bagian anggota gerak yang penting saja untuk digunakan dalam pertandingan Sepak Bola.

Pada Pertandingan Robot Sepak bola, robot juga diharuskan melakukan gerakan-gerakan yang menirukan gerakan manusia seperti berjalan,

berlari, menendang, dan berkoordinasi. Kemampuan robot untuk bekerja dengan cepat dan tepat menjadi penentu utama dalam sebuah pertandingan sepak bola. Salah satu hal yang paling penting dalam pertandingan robot sepak bola adalah kecepatan robot untuk mengejar bola. Dengan memprediksi jarak bola, robot diharapkan bergerak secepat mungkin untuk mengejar bola tersebut. Namun seperti halnya manusia, robot tidak bisa bergerak dari kondisi diam secara langsung dengan kecepatan tinggi ataupun berhenti secara langsung dari kecepatan tinggi. Sehingga diperlukan Akselerasi dan Deselerasi.

Pada saat melakukan akselerasi atau deselerasi kecepatan berjalan, seperti yang dikatakan dalam Hukum I Newton bagian badan robot akan cenderung mempertahankan posisi awal sehingga keseimbangan robot akan terpengaruh. Untuk itu, dalam penelitian ini dilakukan percobaan enambahan kontrol keseimbangan robot saat melakukan pergerakan akselerasi ataupun deselerasi.

1.2 Permasalahan

1. Desain Mekanik Robot Humanoid.
2. Akselerasi dan Deselerasi pada gerakan berjalan robot humanoid.
3. kontrol keseimbangan pada robot humanoid Ichiro saat akselerasi dan deselerasi.

1.3 Tujuan

Tujuan utama dari tugas akhir ini adalah:

1. Merealisasikan desain Mekanik Robot Humanoid.
2. Membuat gerakan akselerasi dan deselerasi pada gerakan berjalan robot humanoid.
3. Mendapatkan kontrol keseimbangan pada robot humanoid Ichiro saat akselerasi dan deselerasi untuk menjaga keseimbangan robot saat melakukan pergerakan akselerasi dan deselerasi.

1.4 Batasan Masalah

Batasan yang digunakan pada penelitian ini adalah platform robot yang digunakan adalah proses akselerasi dan deselerasi serta penambahan kese-

imbangan dilakukan dengan menambahkan algoritma pada platform walking pattern yang sudah ada pada robot Darwin OP

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. Bab I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, penegeasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.

2. Bab II Dasar Teori.

Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini.

3. Bab III Perancangan Sistem dan Impementasi

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah data diolah hingga menghasilkan visualisasi. Guna mendukung itu digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implentasi pada pelaksanaan tugas akhir.

4. Bab IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisanya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati.

5. Bab V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

Halaman ini sengaja dikosongkan.

BAB 2

TINJAUAN PUSTAKA

2.1 Robot *Humanoid*

Robot *humanoid* adalah robot yang penampilan keseluruhannya dibentuk berdasarkan tubuh manusia, mampu melakukan interaksi dengan peralatan maupun lingkungan yang dibuat untuk manusia. Secara umum robot *humanoid* memiliki tubuh dengan kepala, dua buah lengan dan dua kaki, meskipun ada pula beberapa bentuk robot humanoid yang hanya berupa sebagian dari tubuh manusia, misalnya dari pinggang ke atas. Beberapa robot humanoid juga memiliki wajah, lengkap dengan mata dan mulut. Robot humanoid digunakan sebagai alat riset pada beberapa area ilmu pengetahuan. Periset perlu mengetahui struktur dan perilaku tubuh manusia (biomekanik) agar dapat membangun dan mempelajari robot humanoid. Di sisi lain, upaya mensimulasikan tubuh manusia mengarahkan pada pemahaman yang lebih baik mengenai hal tersebut. Kognisi manusia adalah bidang studi yang berfokus kepada bagaimana manusia belajar melalui informasi sensori dalam rangka memperoleh keterampilan persepsi dan motorik. Pengetahuan ini digunakan untuk mengembangkan model komputasi dari perilaku manusia dan hal ini telah berkembang terus sepanjang waktu.

Robot *humanoid* diciptakan untuk meniru beberapa tugas fisik dan mental yang sama seperti manusia menjalani kehidupan setiap harinya. Para ilmuwan dan spesialis dari berbagai bidang termasuk teknik , ilmu kognitif , dan linguistik menggabungkan upaya mereka untuk menciptakan robot yang mirip dengan manusia. Tujuan ilmuwan dan spesialis menciptakan robot *humanoid* adalah agar robot *humanoid* dapat memahami kecerdasan akal manusia dan bertindak layaknya seperti manusia. Jika robot *humanoid* mampu melakukannya, mereka akhirnya bisa bekerja dalam koheisi dengan manusia untuk menciptakan masa depan yang lebih produktif dan berkualitas tinggi. Manfaat lain yang penting untuk mengembangkan robot humanoid adalah untuk memahami tubuh manusia biologis dan proses mental, dari yang sederhana hingga yang berjalan dengan konsep kesadaran dan spiritualitas.

Dalam perencanaan dan pengendalian antara robot *humanoid* dengan robot jenis lain (seperti robot industri) memiliki perbedaan yaitu bahwa gerakan robot harus menyerupai manusia, dengan menggunakan penggerak berkaki, terutama biped kiprah. Perencanaan ideal untuk gerakan robot *humanoid* saat berjalan normal harus menghasilkan konsumsi energi minimum, seperti halnya tubuh manusia. Untuk alasan ini, studi tentang dinamika dan kontrol dari jenis struktur menjadi lebih penting. Untuk menjaga keseimbangan dinamis selama berjalan, robot membutuhkan informasi tentang gaya kontak saat ini dan gerakannya yang diinginkan. Solusi untuk masalah ini bergantung pada konsep utama, *Zero Moment Point (ZMP)*.

Karakteristik lain tentang robot humanoid adalah bahwa mereka bergerak, mengumpulkan informasi (menggunakan sensor) pada "dunia nyata" dan berinteraksi dengan itu, mereka tidak tinggal tetap seperti manipulator pabrik dan robot lain yang bekerja di lingkungan yang sangat terstruktur. Perencanaan dan Pengendalian harus fokus tentang deteksi self-collision, perencanaan jalur dan penghindaran rintangan untuk memungkinkan humanoids untuk bergerak dalam lingkungan yang kompleks.

Ada fitur dalam tubuh manusia yang belum dapat ditemukan di robot *humanoid*. Mereka mencakup struktur dengan fleksibilitas variabel, yang memberikan keselamatan (untuk robot itu sendiri dan kepada orang-orang), dan redundansi gerakan, yaitu lebih derajat kebebasan dan karena itu ketersediaan tugas lebar. Meskipun karakteristik ini diinginkan untuk robot humanoid, mereka akan membawa kerumitan yang lebih dan masalah baru untuk perencanaan dan kontrol.

2.2 Robocup 2016

Pada tahun 2016 *Robocup* diselenggarakan di Leipzig, Jerman. *Robocup* merupakan acara tahunan kontes robot sepak bola tingkat internasional yang diikuti oleh institusi dunia yang memiliki riset dalam bidang robotika. Kontes ini telah diadakan sejak 1997 di Nagoya, Jepang. Cita-cita yang ditonjolkan dari acara ini adalah pada tahun 2050 robot humanoid dapat mengalahkan juara sepak bola FIFA dengan peraturan dari FIFA. Peraturan permainan sepak bola ada di dalam [2].

RoboCup dibagi menjadi beberapa tingkatan sesuai dengan ukuran robot yang dipertandingkan yaitu: *Teen Size*, *Kid Size*, dan *Adult Size*. Pada tugas akhir ini robot yang digunakan termasuk dalam kategori *Kid Size* dengan ukuran tinggi robot antara 40cm – 90cm.

Aturan *RoboCup* berubah setiap waktu menggambarkan perkembangan teknologi yang begitu cepat. Hal tersebut dilakukan untuk medekati liga sepak bola manusia. Para panitia, peneliti dan peserta melakukan evaluasi setiap tahun menyusun aturan-aturan baru. Hal tersebut memberikan tantangan sendiri bagi mereka untuk terus berkembang.

2.3 Kontes Robot Sepak Bola Indonesia (KRSBI) 2017

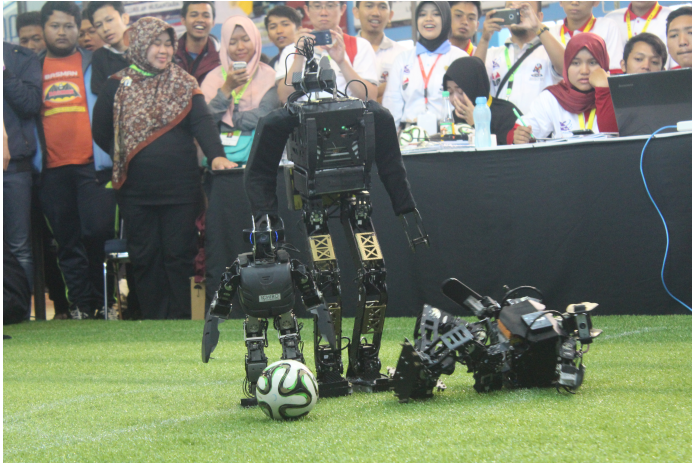
Kontes Robot Indonesia (KRI) adalah kegiatan kontes bidang robotika yang dapat diikuti tim mahasiswa dari institusi atau Perguruan Tinggi Negeri dan Swasta yang terdaftar di Kementerian Riset, Teknologi dan Pendidikan Tinggi.

KRI 2017 diselenggarakan dalam 4 Wilayah Tingkat Regional, yang pembagiannya mencakup sebagai berikut: Regional 1 meliputi wilayah Sumatera dan sekitarnya; Regional 2 meliputi wilayah Jawa bagian barat, Kalimantan (sebagian) dan Sulawesi; Regional 3 meliputi wilayah Jawa bagian tengah, Kalimantan Timur, Kalimantan Selatan; Regional 4 meliputi wilayah Jawa bagian timur, Bali, NTB, NTT, wilayah Maluku dan wilayah Papua. Rangkaian kegiatan KRI Tingkat Regional diselenggarakan mulai 4 Mei 2017 dan berakhir pada tanggal 25 Mei 2017. Pemenang Kontes pada setiap Wilayah Regional akan diundang untuk ikut serta dalam Kontes Robot Indonesia Tingkat Nasional pada tanggal 6-9 Juli 2017.

Untuk dapat mengikuti KRI 2017, setiap tim calon peserta harus mengajukan proposal yang ditujukan ke Panitia Pusat KRI 2017 dengan persetujuan Wakil Rektor/Ketua/Direktur Bidang Kemahasiswaan masing-masing Perguruan Tinggi. Persyaratan tim, mekanisme pengajuan proposal dan form pendaftaran dapat dilihat pada Panduan KRI 2017.

KRSBI adalah Liga Sepak Bola Robot *Humanoid (RoboSoccer Humanoid League)* yang mengacu pada cita-cita organisasi *RoboCup International*, yaitu tahun 2050 mampu mencetak tim sepak bola robot yang mampu melawan tim juara dunia sepak bola. KRSBI 2016 mempertandingkan kelas *KidSize* dengan Rule of the Game diadopsi dari *RoboCup*.

Peserta Kontes Robot Sepak Bola Indonesia berasal dari Perguruan Tinggi dari seluruh Indonesia. Satu tim KRSBI 2017 terdiri dari empat mahasiswa aktif, termasuk mahasiswa program pascasarjana dan 1 (satu) dosen pembimbing. Peserta KRSBI 2016 harus mendaftar terlebih dahulu. Penilaian dilaksanakan sepenuhnya mengacu ada *RoboCup Soccer Humanoid League Rules and Setup 2016* untuk kelas *KidSize*. Peraturan KRSBI 2016 mengacu pada peraturan - peraturan sebelumnya yang ter-



Gambar 2.1 Gambaran pertandingan robot sepak bola *humanoid*

dapat pada [3]. Gambaran pertandingan robot sepak bola *humanoid* dapat dilihat pada gambar 2.1.

2.4 Robot *Humanoid* Darwin-OP

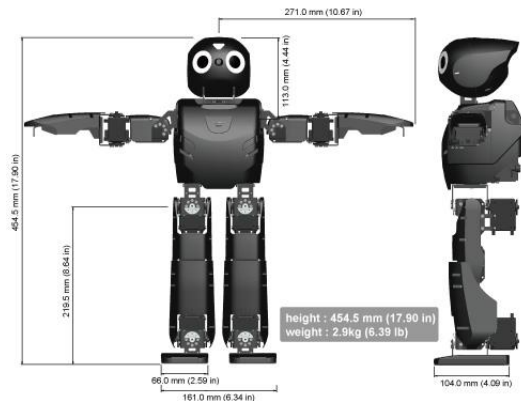
Darwin-OP merupakan kepanjangan dari (*Dynamic Anthropomorphic Robot with IntelligenceOpen Platform*) merupakan robot yang diproduksi oleh Robotis atas kerjasama dengan University Pennsylvania. Gambar 2.4 merupakan tampilan depan dan samping robot Darwin-OP beserta dengan spesifikasi dimensinya. Robot ini sengaja dikembangkan untuk kebutuhan riset dalam permainan sepak bola robot *humanoid* [4] Robot Darwin-OP memiliki perangkat keras yang terdiri atas sebuah PC dengan tipe fitpc2i, modul Wi-Fi, kamera logitech C905, motor *servo* MX-28T dan kontroler CM730. PC pada robot ini berfungsi sebagai pemroses utama dengan sistem operasi berbasis linux. PC ini dapat disambungkan ke jaringan nirkabel dengan modul Wi-Fi. Modul kontroler CM730 merupakan mikrokontroler berbasis ARM yang digunakan untuk mengontrol motor servo MX-28T sebanyak 20 buah. Modul kontroler berkomunikasi dengan PC pemroses utama melalui jalur komunikasi serial. Sensor pada modul kontroler ini yaitu *gyroscope* dan *accelerometer* berfungsi untuk menstabilkan gerakan robot dan mendeteksi jatuh robot.

Informasi mengenai robot ini dibuka untuk publik karena platform ini merupakan *open platform*. Mulai dari informasi mekanik, hingga perangkat lunak yang digunakan. Dengan informasi tersebut, dimungkinkan untuk membuat sebuah Darwin-OP dengan biaya yang lebih murah daripada membeli sebuah robot yang sudah jadi.

2.5 Platform Perangkat Lunak Robot *Humanoid* Darwin-OP [3]

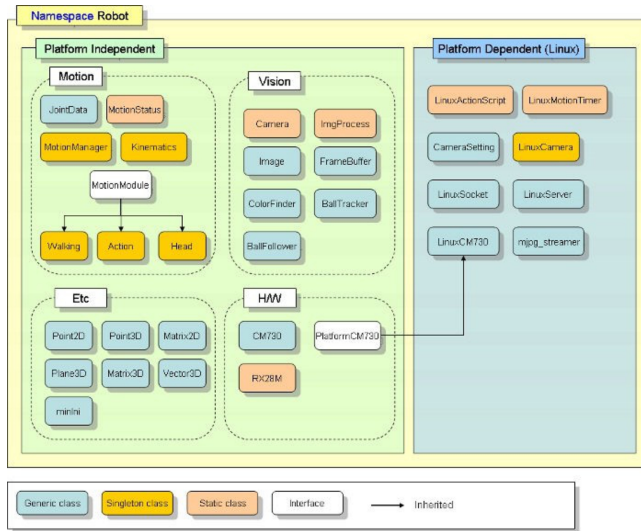
Aspek perangkat lunak DARwIn-OP dibangun dengan kerangka hirarkis dengan mempertimbangkan modularitas dan independensi. Framework DARwIn-OP terdiri dari modul komunikasi perangkat, modul gerak, modul jalan kaki, modul penginderaan, modul perilaku, penglihatan Modul, dan modul diagnostik. Kerangka ini telah dikembangkan dengan bahasa pemrograman C ++ dimana kodenya beroperasi sistem-independen. Aspek sistem operasi yang independen dari kerangka kerja sangat penting sehingga kode tersebut dapat di porting ke sistem operasi komputer yang ada atau yang akan datang, termasuk Sistem Operasi Robot yang baru dikembangkan (ROS).

Pengguna bisa menulis kode perilaku untuk DARwIn-OP tanpa perlu mengembangkan kerangka yang terpisah. Dalam kasus seperti itu,



Gambar 2.2 Tampilan Darwin-OP dan Spesifikasi Dimensi (©Robotis)

Sumber: Manual Darwin-OP



Gambar 2.3 Bagan Framework Darwin

simulator perangkat lunak adalah metode yang paling praktis untuk menulis program dan pengujian kata program untuk DARwIn-OP, mengingat simulator semacam itu memanfaatkan kerangka kerja yang disediakan. Karena sifat open-source DARwIn-OP pengguna dianjurkan untuk berbagai program yang dikembangkan dengan pengguna lain. Namun, pengguna mungkin tidak terbatas pada open-DARwIn-SDK.

Saat ini ada perangkat lunak mandiri lainnya yang tidak terapkan open-DARwIn-SDK. Ada beberapa contoh di mana perangkat lunak telah dikembangkan di "tingkat." Misalnya, pemrograman tingkat rendah dapat menangani sub rutin rutin robot, seperti waktu refresh kamera atau membaca posisi aktuator. Pemrograman tingkat tinggi dapat menangani aspek-aspek yang lebih abstrak dari robot. Tingkat yang berbeda praktis sehingga mengkompilasi ulang keseluruhan kode tidak perlu dilakukan pada perubahan sederhana dalam perilaku atau parameter subrutin. Bagan Framework Darwin dapat dilihat pada Gambar 2.3

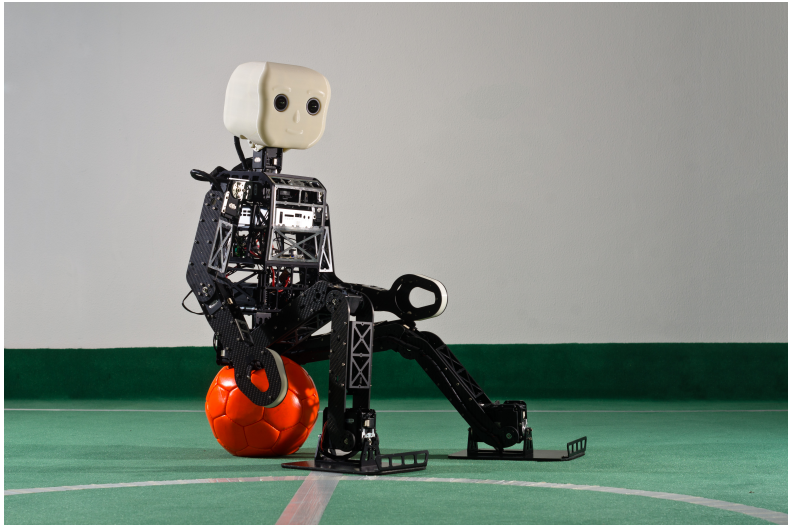
2.6 Robot *Humanoid* Nimbro-OP[6]

NimbRo-OP adalah robot buatan Universitas Bonn Jerman, mempunyai 95 cm dan berat 6,6 kg, Penggunaan servo dibuat seminimal mungkin untuk menjaga desain sesederhana mungkin. Robot ini memiliki 20 derajat kebebasan. *Roll* dan *Pitch* motor di pergelangan kaki, *pitch* motor di lutut, *roll*, *Pitch*, dan motor *yaw* di sendi pinggul, motor *roll* dan *pitch* di bahu, *pitch* motor di siku, dan *yaw* dan *pitch* Motor di leher untuk memberikan gerak pan-miring pada kepala. Sendi dibuat untuk memiliki rentang gerak yang besar, yang sehingga robot lebih fleksibel untuk bergerak.

Semua joint digerakkan oleh aktuator yang dipilih dari Dynamixel MX series diproduksi oleh Robotis. Secara umum, MX-106 digunakan di kaki, dan MX-64 di lengan dan leher. Motor yang lebih lemah dipilih untuk menggerakkan lengan dan leher, untuk mengurangi total biaya dan berat robot. Semua aktuator Dynamixel terhubung dengan satu TTL Bus satu jalur Motor servo, serta semua elektronik lainnya. Komponen ini dapat disupply oleh 14,8 V 3,6 dengan baterai lithium-polymer. [14] Untuk menjaga agar berat tetap rendah, robot ini menggunakan bahan ringan seperti Komposit karbon dan aluminium dan melepas semua bagian tidak diperlukan untuk stabilitas Lengan dan kaki dibuat dari lembaran komposit karbon giling yang terhubung dengan U, bagian aluminium berbentuk dipotong dari lembaran dan ditekuk pada dua sisinya. Batang tubuh, yang menyimpan sebagian besar komponen elektroniknya. Seluruhnya terbuat dari sangkar aluminium. Kepala dan penghubungnya dicetak 3D menggunakan ABS + polimer. Kaki terbuat dari lembaran komposit karbon yang fleksibel. Penendang pada kaki terbuat dari aluminium, sebagian besar persendian dilengkapi dengan bantalan bola di bagian belakang dari aktuator di samping bantalan pada horn servo. Robot NimBro-op dapat dilihat pada gambar 2.4.

NimbRo-OP dilengkapi dengan Zotac Zbox nano XS miniPC, mampu menjalankan operasi berbasis Linux atau Windows Sistem. PC ini dilengkapi prosesor Dual-Core AMD E-450 dengan frekuensi clock 1,65 GHz, memberikan secara signifikan lebih banyak daya komputasi onboard daripada DARwIn-OP. Untuk penyimpanan data, RAM 2 GB (bisa diupgrade sampai 4 GB) dan 64 GB Solid state disk dapat digunakan. Sebuah slot kartu memori juga disediakan. Antarmuka komunikasi yang tersedia adalah USB 3.0, HDMI, dan Gigabit Ethernet. 10.6.

Kepala berisi antena kecil yang merupakan bagian dari USB Adaptor



Gambar 2.4 Tampilan NimBro OP (©Robotis)

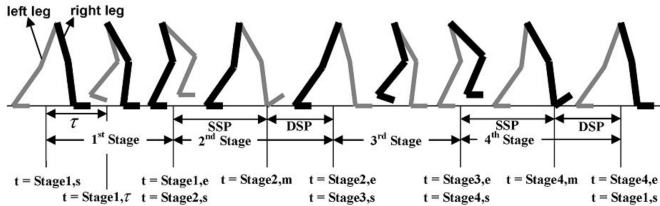
Sumber: Nimbro OP

WiFi, yang mendukung IEEE 802.11b / g / n. Selain PC, papan Robotis CM730 digunakan untuk komunikasi serial frekuensi tinggi dengan Motor servo, papan CM730 dilengkapi dengan Akselerometer sumbu tiga tegra-si dan giroskop tiga sumbu sebagai sensor untuk estimasi sikap. Robot ini menggunakan kamera USB Logitech C905 yang digunakan Robotis di DARwIn-OP. Namun mengganti lensa asli dengan lensa wide-angle custom yang memungkinkan robot untuk memiliki bidang pandang sampai 180.

2.7 Pola Berjalan Robot Humanoid

Metode berjalan static walking adalah metode yang paling umum digunakan dalam robot humanoid. Pola dasar dari metode ini adalah berjalan dengan memposisikan robot sedikit menjongkok. Metode ini banyak digunakan karena memiliki keunggulan dimana kestabilan robot akan lebih terjaga karena mekanisme langkah kakinya cukup kecil dan mengangkat kakinya cenderung rendah. Namun hal tersebut menyebabkan meto-

Stages



Gambar 2.5 Kondisi DDP dan DSP

de berjalan secara *static walking* tampak kaku dan terlihat seperti hanya menggesek-gesek lantai atau surface pijakan. Dengan metode berjalan seperti itu tentunya pergerakan robot akan sangat tergantung dengan *surface* tempat robot berjalan, dimana perbedaan *surface* akan menghasilkan kecepatan berjalan yang berbeda, juga kestabilan robot tentunya akan berbeda saat berada dalam surface yang halus dan kasar. Selain dua hal tersebut, kekurangan dari metode berjalan secara *static walking* adalah langkah robot yang terbatas, dimana sudut langkah yang dihasilkan terlalu kecil untuk bisa berjalan cepat, sehingga untuk mempercepat langkah perlu menaikkan kecepatan aktuator, sehingga metode jalan secara *static walking* kurang efisien. Pola berjalan SSP dan DSP dapat dilihat pada gambar 2.5.

Metode berjalan yang sampai saat ini masih dalam tahap pengembangan untuk memperbaiki mekanisme berjalan *static walking* adalah metode *dynamic walking*, dimana pergerakan robot dibuat menyerupai gerakan berjalan manusia. Yaitu dengan berjalan secara tegap. Dengan metode ini sudut langkah yang dihasilkan saat berjalan lebih besar daripada mekanisme *static walking*. Sudut langkah yang dihasilkan saat berjalan menjadi lebih maksimal sehingga langkah kaki robot bisa dibuat lebih lebar dan lebih tinggi dibandingkan dengan metode *static walking*. Dengan langkah yang lebih lebar tentunya robot akan bergerak lebih jauh di tiap langkahnya sehingga menjadi lebih cepat saat berjalan dengan kecepatan aktuator yang kecil. Kelebihan lain dari penggunaan metode *dynamic wal-*

king adalah kecepatan berjalan robot tidak lagi tergantung pada surface pijakan, karena friksi atau gesekan akan dikurangi dengan mengatur tinggi langkah robot saat berjalan. Kekurangan metode ini yang sampai saat ini masih menjadi kendala adalah kestabilan robot mudah terganggu, dimana langkah yang lebar dan cukup tinggi akan menghasilkan momentum yang membuat kestabilan robot terganggu, untuk itu perlu penelitian lebih lanjut agar robot mampu menjaga keseimbangan saat mengalami gangguan kestabilan. Pola dasar dari metode *dynamic walking* dibuat berdasarkan pola berjalan pada manusia. Pola berjalan dengan kaki tegap, berbeda dengan pola berjalan dari metode *static walking* yang memposisikan robot agak jongkok saat berjalan. Pada pola berjalan *dynamic walking*, tinggi atau panjang kaki akan sangat berpengaruh terhadap jangkauan maksimal yang dapat dicapai dalam setiap langkah, sedangkan pada pola *static walking*, tinggi atau panjang kaki hampir tidak akan berpengaruh terhadap jangkauan langkah saat berjalan. Hal itu disebabkan oleh keterbatasan dari pola *static walking* untuk membentuk sudut langkah yang besar saat berjalan. Kompensasi yang harus di tanggung dalam penggunaan metode *dynamic walking* adalah robot cenderung kurang stabil dan mudah jatuh, untuk itu pengendalian aktuator harus dilakukan secara cermat dengan memperhatikan faktor-faktor lain yang membuat robot kehilangan kestabilan.

2.8 Kontroler PID

Diagram blok sistem kontrol sederhana diberikan oleh gambar 2.6 Kontrol otomatis membandingkan harga yang sebenarnya dari keluaran plant dengan harga yang diinginkan ($r(t)$), menentukan deviasi ($e(t)$), serta menghasilkan sinyal kontrol untuk mengurangi deviasi sampai nol atau sangat kecil. Cara kontroler otomatis bekerja untuk menghasilkan sinyal kontrol disebut aksi pengontrolan



Gambar 2.6 Diagram blok kontroler secara umum

Secara garis besar kontrol otomatis dibagi dua yaitu kontroler konvensional dan kontroler cerdas. Kontroler konvensional terdiri dari kontroler on-off, kontroler Proporsional (P), kontroler Proporsional-Derivatif (PD), kontroler Proporsional-Integral (PI), dan kontroler Proporsional-Integral-Derivatif (PID). Kontroler PID merupakan kontroler yang paling baik di dalam klasifikasi kontroler konvensional. Kontroler cerdas terdiri dari neural network, fuzzy logic, dan algoritma genetika. Algoritma genetika merupakan jenis kontroler yang baik untuk diaplikasikan dalam pencarian solusi sistem dari permasalahan.

Kontroler *PID* (dari singkatan bahasa Inggris: *ProportionalIntegralDerivative controller*) merupakan kontroler mekanisme umpan balik yang biasanya dipakai pada sistem kontrol industri. Sebuah kontroler *PID* secara kontinyu menghitung nilai kesalahan sebagai beda antara setpoint yang diinginkan dan variabel proses terukur. Kontroler mencoba untuk meminimalkan nilai kesalahan setiap waktu dengan penyetelan variabel kontrol, seperti posisi keran kontrol, damper, atau daya pada elemen pemanas, ke nilai baru yang ditentukan oleh jumlahan.

Karena kontroler *PID* hanya mengandalkan variabel proses terukur, bukan pengetahuan mengenai prosesnya, maka dapat secara luas digunakan.[2] Dengan penyesuaian (tuning) ketiga parameter model, kontroler PID dapat memenuhi kebutuhan proses. Respon kontroler dapat dijelaskan dengan bagaimana responnya terhadap kesalahan, besarnya overshoot dari setpoint, dan derajat osilasi sistem. penggunaan algoritma *PID* tidak menjamin kontrol optimum sistem atau bahkan kestabilannya.

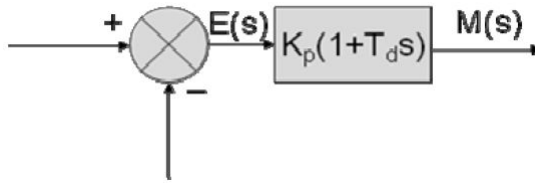
2.8.1 Kontroler Proporsional

Kontroler proporsional (P) merupakan kontroler dengan menggunakan penguatan murni KP. Untuk kontroler proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah dengan $M(s)$ dan $E(s)$ adalah transform Laplace dari $m(t)$ dan $e(t)$, secara berurut.

$$M(t) = K_p E(t) \quad (2.1)$$

$$M(s) = K_p E(s) \quad (2.2)$$

$$\frac{M(s)}{E(s)} = K_p \quad (2.3)$$



Gambar 2.7 Diagram blok kontroler P

Kontroler proporsional memiliki karakteristik: mempercepat proses, Tidak merubah orde proses, meningkatkan overshoot, tidak menghilangkan offset. Diagram blok sistem kontrol P diberikan oleh Gambar 2.7.

2.8.2 Kontroler Proporsional Derivatif

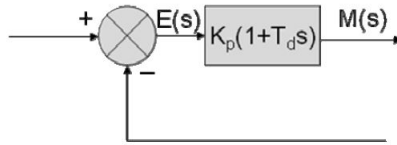
Diagram blok sistem kontrol PD diberikan oleh Gambar 2.8. Dengan KP adalah kepekaan proporsional (penguatan) dan Td adalah waktu derivatif. Keluaran dari kontroler PD sebanding dengan laju perubahan sinyal kesalahan penggerak sehingga aksi kontrol ini sering disebut sebagai (*rate control*). Waktu derivatif Td adalah selang waktu bertambah majunya respon aksi kontrol proporsional yang disebabkan oleh aksi laju²). Kontroler PD tidak mengubah orde proses, tidak menghilangkan osilasi, tidak menghilangkan offset, mengurangi osilasi keluaran, mengurangi overshoot. Aksi pengontrolan PD didefinisikan sebagai didefinisikan dengan persamaan berikut

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.4)$$

$$\frac{M(s)}{E(s)} = K_p (1 + T_d s) \quad (2.5)$$

2.8.3 Kontroler Proporsional Integral

Aksi pengontrolan PI didefinisikan sebagai didefinisikan dengan persamaan berikut :



Gambar 2.8 Diagram blok kontroler PD

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.6)$$

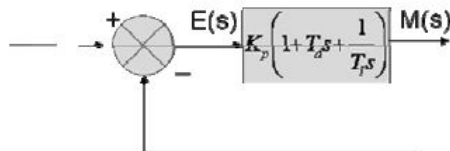
$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (2.7)$$

dengan T_i adalah waktu integral. Kebalikan dari waktu integral T_i disebut laju reset. Laju reset adalah banyaknya pengulangan bagian proporsional dari aksi pengontrolan per menit. Kontroler PI mempunyai karakteristik: menghilangkan offset, mempercepat proses, tetapi menimbulkan osilasi.

2.8.4 Kontroler Proporsional Integral Derivatif

Diagram blok sistem kontrol PID dapat dilihat pada gambar 2.9

Aksi pengontrolan *PI* didefinisikan sebagai didefinisikan dengan persamaan berikut



Gambar 2.9 Diagram blok kontroler *PID*

$$M(s) = K_p \left(1 + T_D s + \frac{1}{T_I s} \right) \quad (2.8)$$

$$m(t) = K_p e(t) + K_p T_D \frac{de(t)}{dt} + \frac{K_p}{T_I} \int_0^t e(t) dt \quad (2.9)$$

$$m(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (2.10)$$

$$M(s) = \left(K_p + K_D s + \frac{K_I}{s} \right) \quad (2.11)$$

Dengan $K_I = \frac{K_p}{T_I}$, dan $K_D = K_p * T_D$ ²⁾.

2.8.5 Metode Tuning PID [5]

2.8.5.1 Metode Ziegler-Nichols

Kontroler *PID* memiliki keunggulan dibandingkan kontroler konvensional jenis lain karena karakteristik dari kontroler *PID* merupakan gabungan dari karakteristik dari kontroler-kontroler yang dimiliki oleh kontroler *P*, kontroler *PD*, dan kontroler *PI*.

Ziegler-Nichols pertama kali memperkenalkan metodenya pada tahun 1942. Metode ini memiliki dua cara yaitu metode osilasi dan kurva reaksi. Kedua metode ditujukan untuk menghasilkan respon sistem dengan lonjakan maksimum sebesar 25

Metode kurva reaksi didasarkan terhadap reaksi sistem loop terbuka. Plant sebagai loop terbuka dikenai sinyal step function. Kalau plant minimal tidak mengandung unsur integrator ataupun pole-pole kompleks, reaksi sistem akan berbentuk *S*. Kelemahan metode ini terletak pada ketidakmampuannya untuk menangani plant integrator maupun plant yang memiliki pole kompleks.

2.8.5.2 Metode Cohen-Coon

Karena tidak semua proses dapat mentolerir keadaan osilasi dengan amplitudo tetap, Cohen-Coon berupaya memperbaiki metode osilasi dengan

menggunakan metode quarter amplitude decay. Respon loop tertutup sistem, pada metode ini, dibuat sehingga respon berbentuk quarter amplitude decay. Quarter amplitude decay didefinisikan sebagai respon transien yang amplitudonya dalam periode pertama memiliki perbandingan sebesar seperempat ($1/4$).

Kontroler proportional K_p ditala hingga diperoleh tanggapan quarter amplitude decay, periode pada saat tanggapan ini disebut T_p dan parameter T_i dan T_d dihitung dari hubungan K_p dengan T_p . Sedangkan penalaan parameter kontroler PID adalah sama dengan yang digunakan pada metode Ziegler-Nichols. Selain cara tersebut, metode Cohen Coon ini bisa dihitung dengan aturan praktis yang parameter parameter plant-nya diambil dari kurva reaksi.

2.8.5.3 Metode Trial and Error

Pada suatu sistem yang tidak dapat diketahui parameter-parameter untuk menggunakan metode-metode tuning PID diatas, maka dapat digunakan metode *Trial and Error*. *Trial and Error* metode untuk mencapai sebuah tujuan melalui berbagai macam cara. Upaya ini yang dilakukan tersebut dilakukan beberapa kali hingga akhirnya mendapatkan cara yang paling sesuai. Kesalahan atau kekiliruan dicatat untuk dievaluasi dan sebagai bahan pembelajaran. Upaya tersebut dilakukan melalui lebih dari satu cara hingga satu cara dapat berhasil. *Trial and Error* dapat juga didefinisikan sebagai sebuah metode demi mencari sebuah solusi yang benar dan memuaskan melalui berbagai macam cara dan teori hingga akhirnya kesalahan dapat dikurangi atau dihilangkan sama sekali. Hal ini juga kadang berarti melibatkan eksperimen praktis dan pengalaman daripada teori.

2.9 Sistem Operasi Lubuntu

Lubuntu merupakan sistem operasi adalah sebuah proyek yang dimaksudkan untuk menghasilkan suatu turunan resmi dari sistem operasi Ubuntu yang "lebih ringan, lebih sedikit menggunakan sumber daya dan efisien energi", menggunakan lingkungan desktop LXDE. Desktop LXDE menggunakan window manager Openbox dan dimaksudkan untuk menjadi sistem yang rendah persyaratan, menggunakan RAM sedikit untuk netbook, mobile devices dan PC (komputer) tua. Dalam tugas ini Lubuntu akan bersaing dengan Xubuntu. Nama Lubuntu adalah kombinasi dari LXDE dan Ubuntu. LXDE adalah singkatan dari Lightweight X11 Desk-



Gambar 2.10 Logo sistem operasi lubuntu

top Environment. Desktop LXDE pertama kali disediakan sebagai paket opsional untuk Ubuntu sejak versi 8.10 "Intrepid Ibex". LXDE juga dapat di-instal pada Ubuntu versi-versi sebelumnya. Pada bulan Februari 2009 Mark Shuttleworth mengundang proyek LXDE menjadi proyek tersendiri dalam komunitas Ubuntu, dengan tujuan khusus untuk menghasilkan turunan resmi Ubuntu baru yang dinamakan Lubuntu. Pada bulan Maret 2009, proyek Lubuntu dimulai pada Launchpad oleh Mario Behling, termasuk peresmian logo awal. Tujuan proyek Lubuntu adalah untuk mendapatkan sokongan dana dari [Canonical]]. Proyek ini juga membuat sebuah halaman proyek wiki Ubuntu resmi, juga dipimpin Behling, yang mencakup daftar aplikasi, paket dan komponen. Pada bulan Agustus 2009, test ISO pertama diterbitkan sebagai Live CD, tanpa opsi untuk penginstalan. Logo dari sistem operasi Lubuntu dapat dilihat pada gambar 2.10.

2.10 Kontroler CM 730

Kontroler CM730 adalah kontroler yang dirancang untuk mengontrol banyak servo. Kontroller ini menggunakan prosesor ARM *Cortex F103* 32-bit CPU (clocked @ 72 MHz) (512 KB Flash, 64KB SRAM), dilengkapi dengan sensor giroskop 3-axis, akselerometer 3-axis, dan sensor tegangan input. Mempunyai interface 5 x LED , 2 x RGB LED, 3 x Button, 1 x Buzzer, serta mempunyai tambahan 3 x ADC /I/O Port. Kontroler ini dilengkapi dengan USB Port, Serial Port , 5x TTL Ports untuk komunikasi dengan kontroler lain atau aktuator. kontroler ini bisa mengontrol sampai 255 buah servo jenis dynamixel. Kontroler ini mempunyai berat 51g, dengan dimensi 80mm x 7.5mm x 20mm. Input voltage 8V 16.8V dan konsumsi arus maksimum 150mA, konsumsi arus standby 50mA. De-

ngan temperature operasi -65 derajat celcius hingga 80 derajat celcius. Kontroler CM 730 dapat dilihat pada gambar 2.11



Gambar 2.11 Kontroler CM 740

Halaman ini sengaja dikosongkan.

BAB 3

PERANCANGAN SISTEM DAN IMPLEMENTASI DESAIN

Dalam bab ini akan dijelaskan tentang desain, perancangan sistem dan implementasi secara keseluruhan, mulai dari desain mekanik robot, desain elektronik sistem keseimbangan robot, desain sistem keseimbangan pada akselerasi dan deselerasi robot, implementasi desain mekanik, hingga implementasi akselerasi dan deselerasi pada gerakan berjalan robot. Perangkat keras yang digunakan adalah satu buah unit robot humanoid Ithaaro yang mempunyai 20 derajat kebebasan dengan Intel NUC5i5RY sebagai komputer utama dan mikrokontroler CM730 sebagai kontroler aktuator robot. Dalam sistem ini, semua data dari sensor diproses oleh komputer utama dengan sistem operasi Linux sebagai pengolah algoritma berjalan dan mengirimkannya ke mikrokontroler cm730 untuk mengirimkan perintah sudut ke servo.

3.1 Desain Mekanik Robot

Perancangan mekanik robot Ithaaro bertujuan untuk mengikuti pertandingan robot sepak bola robocup kategori kids size dan teen size, dengan dasar referensi desain dari robot Nimbro Open Platform buatan Universitas Bonn, Jerman. Perubahan utama yang dilakukan adalah proses resize karena robot Nimbro OP adalah robot teen size dengan ukuran 95 cm, sementara batas maksimal robot kids size adalah 90 cm. Proses resize dilakukan dengan cara mengurangi jarak antar join aktuator yang arahnya vertikal dengan skala 80% dari nilai referensi 95 cm sehingga didapatkan hasil tinggi robot 84,39 cm. Dengan tinggi 84,39 cm robot ini dapat mengikuti kategori kids size dengan batas maksimal tinggi 90 cm, tetapi masih dapat mengikuti pertandingan di kategori teen size dengan batas tinggi minimal 70 cm. Data ukuran robot dapat dilihat pada tabel 3.1.

Namun, selain tinggi ada batas - batas dimensi lain yang harus diperhatikan pada setiap kategori untuk menjaga proporsionalitas robot. Batas - batas tersebut diantaranya adalah ukuran lebar tubuh ketika tangan tidak mengembang, ukuran lebar tubuh ketika tangan mengembang,

Tabel 3.1 Tabel nilai dimensi hasil proses desain robot

No	Deskripsi	Status	
		Ukuran, Max (cm)	Ukuran, Robot
1	Tinggi Robot (HTop) diukur dari kaki ke kepala. HTop harus memenuhi, $40\text{ cm} \leq \text{HTop} \leq 90\text{ cm}$	90	88
2	Hcom (Tinggi posisi Center of Gravity diukur dari tapak kaki)	-	44
3	Ukuran luas telapak kaki (K). Maksimal $= (2.2 \times \text{Hcom}^2)/32$	292.282	228
4	Ukuran lebar tubuh saat tangan tidak mengembang. Maksimal $= 0.55 \times \text{HTop}$	48.4	29
5	Ukuran lebar tubuh saat tangan mengembang. Maksimal $= 1.2 \times \text{HTop}$	105.6	95

dan luas maksimal telapak kaki. setiap kriteria tersebut mempunyai perhitungan sendiri berdasarkan nilai tinggi Center of Mass (HCoM) robot dan tinggi robot (HTop). Pada kriteria lebar tubuh menggunakan acuan tinggi robot (HTop) dengan perhitungan $0.55 \times \text{HTop}$ pada lebar tubuh saat tidak mengembang, dan $1.2 \times \text{HTop}$ pada lebar tubuh saat mengembang. Pada kriteria luas tapak kaki maksimal dihitung berdasarkan nilai $(2.2 \times \text{HCoM}^2)/32$.

Dalam kriteria-kriteria tambahan tersebut robot Ithaaro perlu penyesuaian terhadap lebar robot saat tangan direntangkan dan luas maksimal telapak kaki. Ukuran lebar tubuh saat mengembang dikurangi dengan mengurangi panjang lengan, sementara luas alas kaki dikurangi dengan cara mengurangi luasan dengan tetap mempertahankan rasio panjang x lebar. Panjang x lebar alas kaki dikurangi dengan skala 8:10 terhadap nilai awal. Karena untuk pengukuran alas kaki membutuhkan nilai HCoM, sedangkan HCoM baru bisa diukur setelah robot jadi, maka ditetapkan nilai HCoM pada robot setinggi 40cm dari tanah. Nilai dimensi hasil proses desain robot dapat dilihat pada tabel 3.1. Visualisasi perbandingan antara

Tabel 3.2 Spesifikasi aktuator dengan input tegangan 14.8V

Aktuator	Berat (g)	Torsi (kg.cm)	Kecepatan (RPM)
MX 28	72	31.6	67
MX 64	126	74	78
MX 106	153	102	55

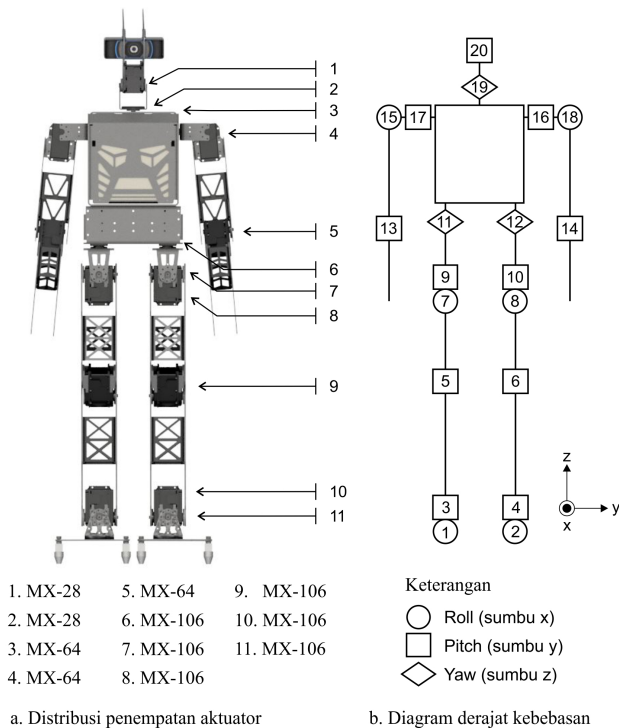
robot Ithaaro dan Nimbro dapat dilihat seperti pada gambar 3.1.

Robot Ithaaro memiliki 20 DoF dapat dilihat pada gambar 3.1, 12 DoF untuk kaki, 6 DoF untuk tangan, dan 2 DoF untuk kepala. Tinggi total robot adalah 88 cm. Rangka robot dibuat menggunakan aluminium lembaran yang dipotong menggunakan mesin CNC laser. Pada bagian kaki dan pinggul menggunakan aluminium dengan tebal 3mm, sedangkan pada bagian badan, tangan, dan kepala menggunakan aluminium dengan tebal 2mm. Semua aluminium yang digunakan menggunakan aluminium tipe 5053. Aktuator yang digunakan pada robot ini adalah Dynamixel MX-106, MX-64, dan MX-28. Spesifikasi aktuator dapat dilihat pada tabel 3.2.

MX-106 digunakan pada 12 DoF di kaki, MX-64 digunakan pada 6 DoF di tangan, sedangkan MX-28 digunakan pada 2 DoF di kepala. Pemilihan motor servo tersebut digunakan sesuai kebutuhan torsi. Pada kaki digunakan motor servo MX-106 yang mempunyai torsi 106 kg/cm karena merupakan bagian yang menopang bagian tubuh lain. Visualisasi distribusi aktuator pada robot Ithaaro dapat dilihat pada gambar 3.1.

3.1.1 Kepala

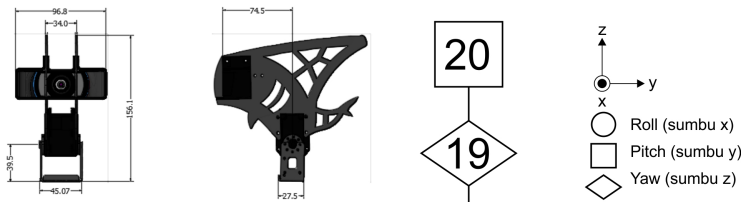
Desain kepala adalah modifikasi dari robot Nimbro dengan memperpanjang leher dan menghilangkan bagian aluminium dibawah kepala dengan tujuan agar pandangan kamera dapat menjangkau daerah bawah depan dan samping jika bola berada tepat didepan atau disamping kaki. Kepala robot mempunyai dua Derajat kebebasan untuk gerakan pitch dan yaw. Gerakan ini mirip gerakan manusia saat menggeleng dan mengangguk. Pada kepala terdapat dua aktuator MX 28 dan satu buah sensor berupa kamera. Kepala robot dibuat seringan mungkin agar tidak terlalu membebani kerja aktuator. Desain, ukuran, dan derajat kebebasan kepala robot dapat dilihat pada gambar 3.2.



Gambar 3.1 Distribusi aktuator dan diagram derajat kebebasan robot Ithaaro

3.1.2 Badan

Badan berfungsi sebagai wadah dan pelindung perangkat elektronik dari gangguan luar dan saat robot jatuh. Komponen elektronik yang terdapat pada badan diantaranya; CPU, kontroler, kompas, power supply distributor, dan baterai. Selain itu, pada badan terdapat dua buah aktuator pitch untuk bahu, satu aktuator yaw untuk kepala, dan dua aktuator yaw untuk kaki. Perubahan yang dilakukan dari robot Nimbro adalah motif profil aluminium pada badan lebih disesuaikan pada kebutuhan karena terdapat daerah yang rawan bengkok ketika jatuh. Daerah tersebut berada di sekitar lengan dan kepala. Pada daerah-daerah tersebut profil aluminium dikurangi atau dihilangkan untuk memperkokoh badan. Selain itu pada

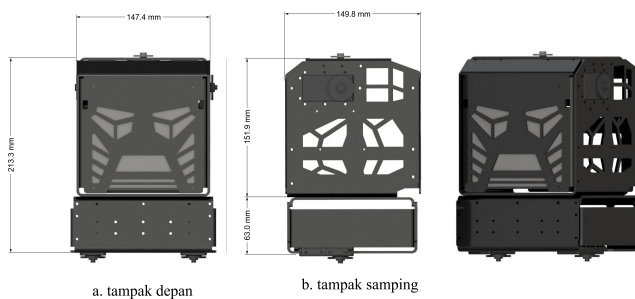


Gambar 3.2 Ukuran dan derajat kebebasan kepala robot

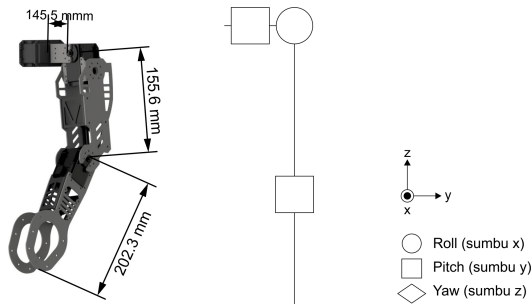
badan bagian depan juga ditambahkan sistem suspensi untuk meminimalisir dampak gaya yang diterima robot saat jatuh kedepan. Sistem suspensi tersebut menggunakan shockbreaker mobil rc dan nilon sebagai suspensi utamanya. Desain dan ukuran badan dapat dilihat pada gambar 3.3.

3.1.3 Lengan

Desain mekanik lengan robot mempunyai tujuan utama untuk gerakan bangun dan melempar bola. Oleh karena beban berat mengangkat badan saat gerakan bangun, aktuator yang digunakan pada lengan robot adalah MX 64 yang mempunyai torsi 74kg.cm pada input tegangan maksimal 14.8V. Perubahan yang dilakukan dari robot Nimbro adalah memperpendek panjang tangan untuk menyesuaikan proporsional dengan tinggi robot. Selain itu dilakukan modifikasi pada bagian ujung tangan agar lebih kokoh saat menopang tubuh pada gerakan bangun. Tangan robot memiliki 3 Dof,



Gambar 3.3 Desain dan ukuran badan robot



Gambar 3.4 Desain mekanik dan diagram derajat kebebasan lengan robot

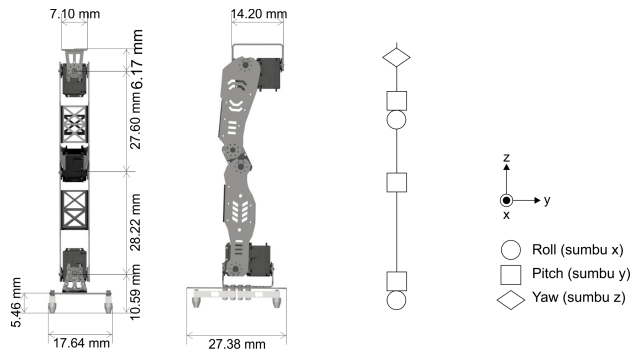
dua dof pada bahu untuk gerakan pitch dan roll, serta satu dof pada siku untuk gerakan pitch. Desain mekanik lengan robot dan diagram derajat kebebasan dapat dilihat pada gambar 3.4.

3.1.4 Kaki

Desain mekanik kaki robot bertujuan untuk menopang badan robot pada saat berdiri, meniru gerakan berjalan manusia, dan menendang bola. Aktuator yang digunakan pada kaki robot adalah MX 106 yang mempunyai torsi 102kg.cm pada tegangan input maksimal 14.8V. Perubahan yang dilakukan dari robot Nimbro adalah memperpendek panjang kaki untuk menyesuaikan dimensi peraturan lomba KRI dan Robocup, serta mengubah bentuk kaki agar lebih terlihat proporsional. Setiap kaki memiliki enam derajat kebebasan, tiga derajat kebebasan pada pangkal paha untuk gerakan yaw, pitch, dan roll, satu derajat kebebasan pada lutut untuk gerakan pitch, dan dua derajat kebebasan pada pergelangan kaki untuk gerakan pitch dan roll. Desain mekanik kaki robot dan diagram derajat kebebasan dapat dilihat pada gambar 3.5.

3.1.5 Telapak Kaki

Desain Telapak kaki pada robot Ithaaro dibuat khusus untuk dapat berjalan di rumput. Perubahan yang dilakukan terhadap robot Nimbro adalah diperkecilnya luas telapak kaki serta penambahan load cell dan pull. Telapak kaki robot Ithaaro mempunyai empat loadcell yang disusun di bagian kanan dan kiri setiap telapak kaki. Dua loadcell sejajar di bagian

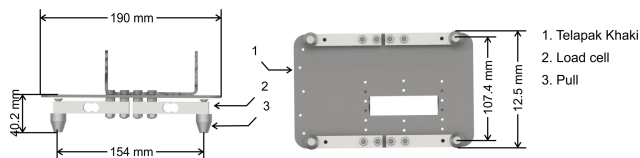


Gambar 3.5 Desain mekanik dan diagram derajat kebebasan kaki robot

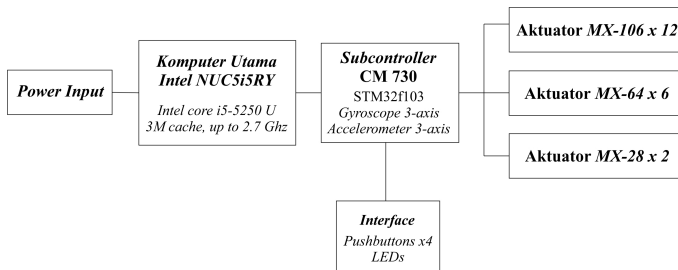
kanan paralel dengan dua loadcell yang lain di sebelah kiri. Diujung setiap loadcell terdapat pull seperti pada sepatu pemain sepak bola. Tujuan ditambahkannya loadcell adalah untuk keseimbangan tekanan robot saat berdiri, sedangkan penambahan pull adalah untuk menabahka grip pada saat robot berjalan maupun menendang bola. Desain mekanik telapak kaki robot dapat dilihat pada gambar 3.6.

3.2 Perancangan *Hardware* Elektronik Sistem

Sensor untuk sistem keseimbangan robot menggunakan accelerometer dan gyrometer yang tertanam dalam kontroler CM730. Data dari accelerometer dan gyrometer dapat diakuisisi langsung oleh cpu melalui komunikasi serial dengan kontroler cm730. Sensor IMU pada robot digunakan untuk keseimbangan berjalan, dan juga digunakan sebagai parameter kondisi robot apakah sedang jatuh atau tidak. Bagan sistem elektronik dapat dilihat pada gambar 3.7 .



Gambar 3.6 Desain mekanik dan telapak kaki robot



Gambar 3.7 Bagan sistem elektronik robot

3.3 Perancangan Software Sistem Keseimbangan pada Akselerasi dan Deselerasi Robot

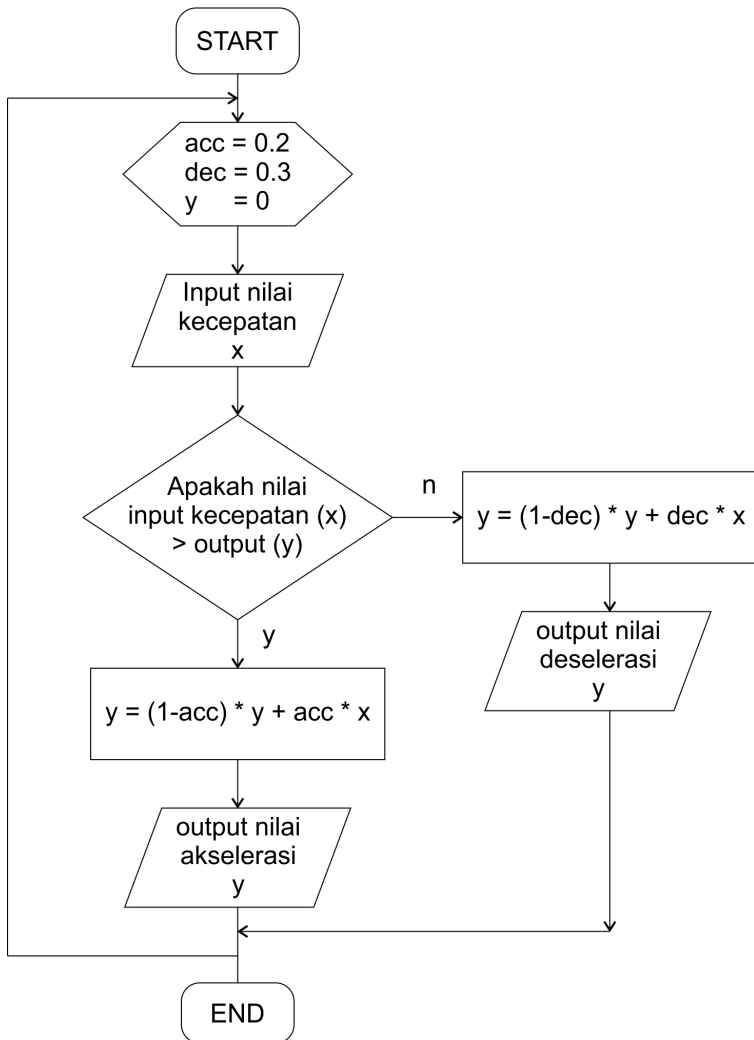
Dalam Sistem keseimbangan pada akselerasi dan deselerasi robot, sistem keseimbangan tidak berhubungan langsung dengan akselerasi dan deselerasi, melainkan sistem keseimbangan diterapkan setelah diterapkannya sistem akselerasi dan deselerasi robot sehingga jika sistem keseimbangan diterapkan pada saat robot berjalan tanpa akselerasi dan deselerasi perlu dilakukan penyesuaian ulang parameter kontrol.

Dalam tugas akhir ini tidak dilakukan perancangan pembangkitan gaya berjalan dari nol, melainkan memodifikasi gaya berjalan dari framework darwin OP dengan menambahkan proses akselerasi-deselerasi dan kontrol keseimbangannya.

3.3.1 Akselerasi dan Deselerasi Robot

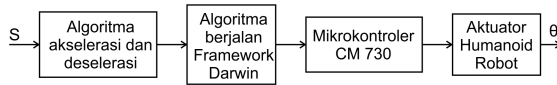
Akselerasi dan deselerasi pada robot berfungsi ketika robot berjalan menuju tsrget tertentu. Dengan adanya akselerasi, diharapkan robot dapat lebih seimbang saat berjalan mendekati target. penambahan algoritma akselerasi pada robot dilakukan pada proses input kecepatan berjalan. algoritma akselerasi dan deselerasi pada program robot berupa menambahkan proses inkremental dari parameter kecepatan algoritma berjalan robot yang sudah ada pada framework darwin. Diagram blok akselerasi dan deselerasi dapat dilihat pada gambar 3.9.

Algoritma sistem akselerasi menggunakan sstem penambahan nilai setiap *loop* program. Kecepatan yang dimasukkan tidak langsung diterapkan pada robot melainkan, disimpan dulu sebagai batas incremen-



Gambar 3.8 Flowchart algoritma akselerasi dan deselerasi robot

tal. Flowchart algoritma akselerasi dan deselerasi dapat dilihat pada gambar 3.8.



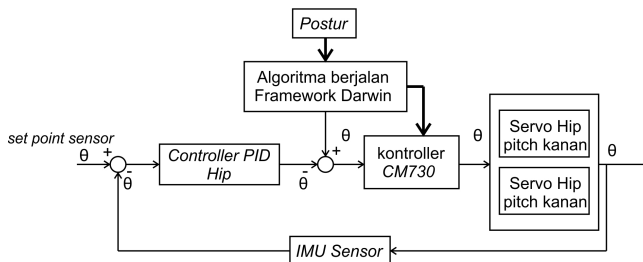
Gambar 3.9 Diagram blok akselerasi dan deselerasi robot

3.3.2 Sistem Keseimbangan Robot

Sistem keseimbangan yang ditambahkan pada algoritma robot adalah spesifik pada kontrol aktuator bagian *hip* dan tangan robot bagian bahu robot. Terdapat dua buah aktuator bagian hip robot yang akan dikontrol. Sedangkan pada tangan robot terdapat dua buah aktuator juga di bagian bahu. Dengan menambahkan keseimbangan pada hip dapat membuat robot lebih stabil saat berjalan.

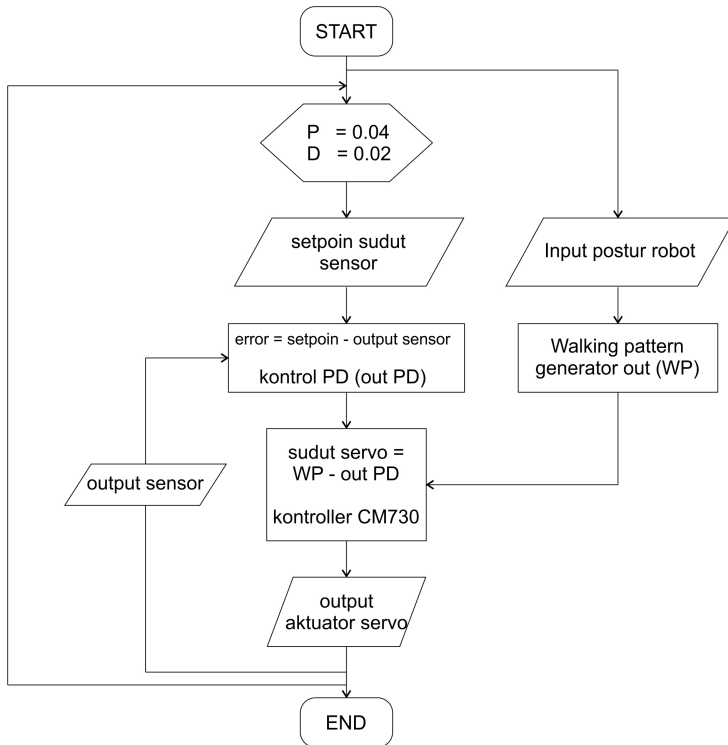
3.3.3 Kontroler PD pada bagian *Hip* robot

Dasar dari pemilihan bagian *Hip* sebagai *plant* adalah bagian *hip* merupakan bagian penghubung antara kaki dan badan, sehingga bagian ini dapat berperan secara langsung terhadap kestabilan badan ke arah depan dan belakang saat robot dalam kondisi diam maupun berjalan. Kontroler yang digunakan adalah kontroler PD karena membutuhkan respon cepat untuk menuju *steady state*. Pada kontroler PD, berdasarkan hasil percobaan *trial and error* diketahui nilai yang paling bagus adalah dengan P bernilai 0.04 dan D bernilai 0.02. Diagram blok Sistem keseimbangan robot dengan kontrol PD pada hip dapat dilihat pada gambar 3.10. Sedangkan



Gambar 3.10 Diagram blok sistem keseimbangan robot dengan kontrol PD pada bagian *hip pitch*

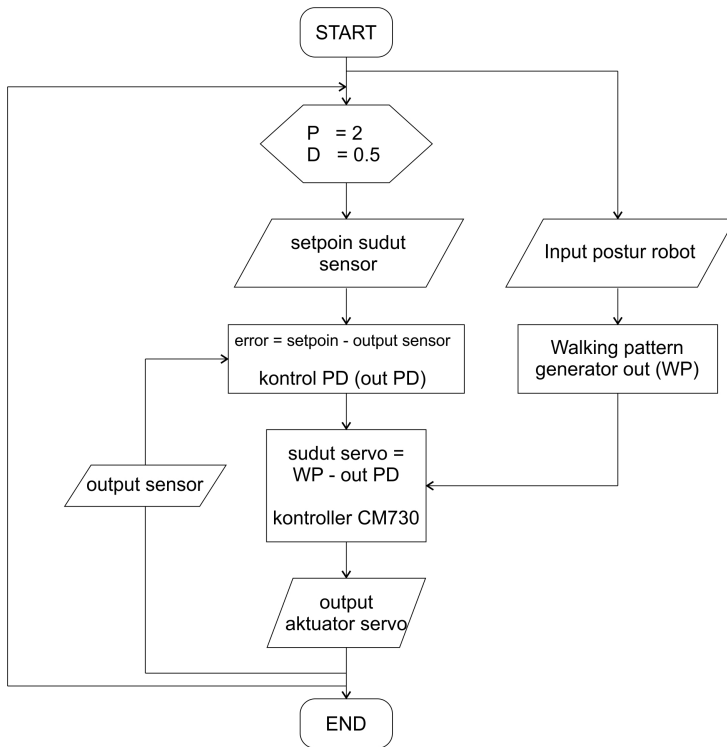
Flowchartnya dapat dilihat pada gambar 3.11.



Gambar 3.11 Flowchart sistem keseimbangan robot

3.3.4 Kontroler PD pada lengan robot

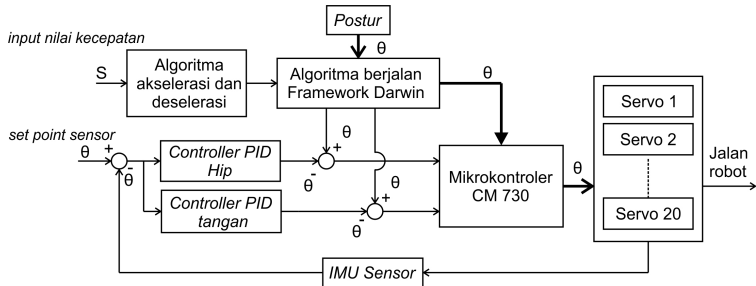
Kontroler PD pada lengan robot diterapkan karena ayunan lengan secara tidak langsung dapat mempengaruhi keseimbangan badan robot. Kontroler yang digunakan adalah kontroler PD karena membutuhkan respon cepat untuk menuju *steady state*. Pada kontroler PD, berdasarkan hasil percobaan *trial and error* diketahui nilai yang paling bagus adalah dengan P bernilai 2 dan D bernilai 0.5. Flowchart Sistem keseimbangan robot dengan kontrol PD pada lengan dapat dilihat pada gambar 3.12.



Gambar 3.12 Flowchart sistem keseimbangan tangan robot

3.3.5 Sistem Keseimbangan pada Akelerasi dan Deselerasi Robot

Sistem keseimbangan robot terpisah dari proses akselerasi dan deselerasi, namun proses akselerasi dan deselerasi mempengaruhi pengaturan parameter kontrol pada sistem keseimbangan robot. Sistem akselerasi ditambahkan sebelum algoritma berjalan robot darwin sehingga mempengaruhi *walking pattern robot*. Setelah *walking pattern* diproduksi, kontrol sistem keseimbangan dikurangkan dari data sudut *walking pattern*. Kemudian kontroller CM730 membuat trajektori planning dari *walking pattern* sehingga dapat digunakan untuk menjalankan aktuator. Diagram blok Sistem keseimbangan robot dapat dilihat pada gambar 3.13.



Gambar 3.13 Diagram blok sistem keseimbangan robot dengan akselerasi

3.4 Implementasi Desain Mekanik

Setelah proses desain selesai, dilakukan proses fabrikasi part aluminium menggunakan mesin *laser cutting* dan mesin *bending* untuk mendapatkan bentuk yang diinginkan. proses fabrikasi part dilakukan di Perusahaan Dempo Metal Laserindo, Rungkut, Surabaya. Part-part hasil fabrikasi kemudian dirakit menjadi robot. Namun sebelum perakitan part-part tersebut harus dipastikan dulu kesesuaian bentuk dan keseimbangan berat antara bagian kanan dan kiri robot. Kondisi robot sebelum perakitan dapat dilihat pada gambar 3.14.



Gambar 3.14 Kondisi part robot sebelum dirakit

3.4.1 Kondisi Sebelum Perakitan

Kondisi part aluminium sebelum perakitan perlu dibersihkan dan ditimbang dulu untuk memastikan bagian kanan dan kiri robot seimbang, jika tidak seimbang maka perlu ada yang dikurangi dengan cara dikikir. Setelah penimbangan, diketahui bahwa part aluminium robot sudah seimbang antara bagian kanan dan kirinya. Tabel rincian berat part robot dapat dilihat pada tabel 3.3.

Pada gambar 3.14 terlihat bahwa robot sudah dipasang per bagian namun belum dirakit secara utuh. Proses yang dilakukan sebelum perakitan adalah pembersihan aluminium dengan *thinner* dan pengamplasan secara halus part aluminium robot agar setelah dicat aluminium tidak mudah mengelupas.

Tabel 3.3 Rincian berat part robot sebelum dirakit

no	nama	Berat pengukuran ke - (gram)			Rata - rata (gram)	Additional fixed weight	total
		I	II	III			
1	Bagian Kiri	824	824	827	826		1926
2	Fore arm	72	72	72	72	140	212
3	Upper arm	92	93	93	93	140	233
4	Shoulder	19	20	20	20		20
5	Thigh	198	198	197	198	328	526
6	Shank	184	184	184	184	164	348
7	Lower Hip	49	49	49	49		49
8	Foot	210	210	210	210	328	538
	Bagian Kanan	821	820	821	821		1921
9	Fore arm	72	72	72	72	140	212
10	Upper arm	92	92	92	92	140	232
11	Shoulder	16	16	16	16		16
12	Thigh	198	197	198	198	328	526
13	Shank	184	184	184	184	164	348
14	Lower Hip	49	49	49	49		49
15	Foot	210	210	210	210	328	538
	Badan	666	664	666	665		2341
16	Head	117	117	118	117	232	349
17	Neck	14	13	13	13		13
18	Body	316	316	316	316	780	1096
19	Hip	219	218	213	219	664	883
						Total	6188

3.4.2 Kondisi Setelah Perakitan

Setelah setiap bagian dirakit, bagian-bagian robot ditimbang lagi untuk memastikan bagian kanan dan kiri robot seimbang. Kondisi robot setelah perakitan dapat dilihat pada gambar 3.15.

Proses perakitan robot memerlukan waktu tujuh hari karena banyak part yang harus disesuaikan dengan aktuator dan peletakan sistem elektroniknya. Hal yang dilakukan setelah proses perakitan adalah percobaan berdiri dan duduk. dapat dilihat pada gambar 3.15 robot difoto saat keadaan berdiri. Robot difoto dari samping sehingga terlihat postur berdiri yang kakinya agak menekuk. Hal ini dilakukan untuk menyesuaikan ke-setimbangan robot saat berjalan agar aktuator kaki tidak cepat panas saat robot berjalan. Diketahui setelah pengukuran CoM robot adalah 40 cm.

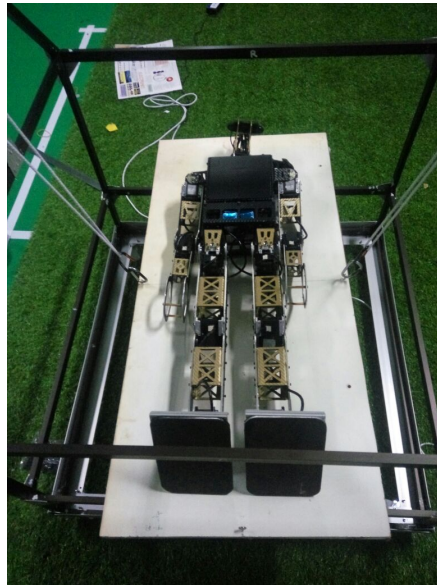


Gambar 3.15 Kondisi part robot setelah dirakit

3.4.3 Pengukuran CoM

Pengukuran tinggi *center of mass* (com) diperlukan untuk mengetahui nilai kriteria luas alas kaki apakah melebihi batas atau tidak, jika ternyata luas alas kaki melebihi batas peraturan robocup, maka perlu pengaturan ulang distribusi komponen agar posisi tinggi *center of mass* sesuai dengan desain.

Pengukuran *center of mass* dilakukan setelah robot dirakit dengan cara menggunakan papan yang diakitkan dengan tali pada titik tengah sisinya kemudian digantungkan. Cara pengukurannya adalah robot diletakkan diatas papan kemudian dicari titik setimbangnya dengan cara menggeser robot. Nilai *CoM* adalah panjang dari titik tengah papan ke ujung robot. Pengukuran ini dapat digunakan untuk mencari *CoM* pada sumbu Z dan Y robot. Berdasarkan pengukuran didapatkan nilai pusat massa terhadap sumbu Z sebesar 43 cm. Pengujian nilai *CoM* dapat dilihat pada gambar 3.16.



Gambar 3.16 proses pengukuran tinggi pusat massa

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini akan dijabarkan pengujian hasil sistem yang telah didesain meliputi pengujian desain mekanik robot, pengujian sistem elektronik robot, kemudian sistem akselerasi dan deselerasi pada robot, sistem keseimbangan berjalan robot, dan sistem keseimbangan robot pada proses akselerasi dan deselerasi pada robot.

4.1 Pengujian Mekanik

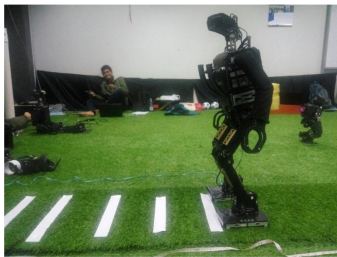
Pengujian mekanik bertujuan untuk menguji performa robot dalam gerakan dasar pada robot sepak bola humanoid. Pengujian ini penting karena robot sepak bola humanoid memerlukan performa yang tinggi dalam hal berjalan agar dapat bersaing dengan robot lawan, seperti halnya pemain sepak bola pada manusia yang harus memiliki fisik yang lebih kuat daripada manusia bukan pemain sepak bola.

Robot diuji performanya dengan tujuan untuk mengetahui batasan kemampuannya, hal yang diuji pada pengujian mekanik diantaranya gerakan berjalan lurus, gerakan berjalan memutar, dan gerakan berjalan menyamping. Gerakan berjalan lurus merupakan gerakan dasar pada robot humanoid, pada robot sepak bola berungsi untuk mengejar bola dan menggiring bola. Sedangkan gerakan berjalan memutar digunakan untuk mengarahkan robot kearah bola atau gawang. Sementara gerakan menyamping merupakan alternatif gerakan memutar untuk mengarahkan robot kearah bla atau gawang.

Pengujian desain mekanik robot dilakukan dengan cara melakukan tiga gerakan dasar robot humanoid, berjalan lurus, berputar, dan berjalan kesamping yang dilakukan di rumput sintetis. Hasil pengujian ini akan dibandingkan dengan robot Darwin OP. Pemilihan pembeding Darwin OP adalah karena menggunakan platform yang sama, dalam kategori yang sama, yaitu kids size, dan juga robot Darwin OP merupakan robot pabrik-an yang sudah teruji performanya.

4.1.1 Gerakan Berjalan Lurus

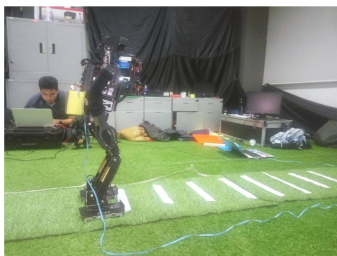
Pengujian mekanik dengan gerakan berjalan lurus digunakan untuk mengetahui keseimbangan pitch mekanik robot. Pengujian menggunakan walking pattern default dari platform darwin dengan pose yang sudah ditentukan. Pengambilan data dilakukan dengan menjalankan robot pada lintasan sepanjang 3 meter yang mempunyai meteran dengan titik nol adalah titik start robot. Proses pengujian dapat dilihat pada gambar 4.1, pengukuran dilakukan dengan cara menghitung waktu yang diperlukan oleh robot menyelesaikan gerakan berjalan pada akhir lintasan, kemudian dihitung kecepatan dalam satuan cm perdetik. Berdasarkan hasil pengujian didapatkan bahwa kecepatan berjalan robot adalah 23cm perdetik.



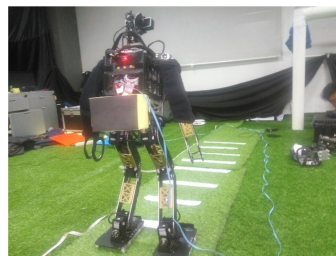
a



b



c

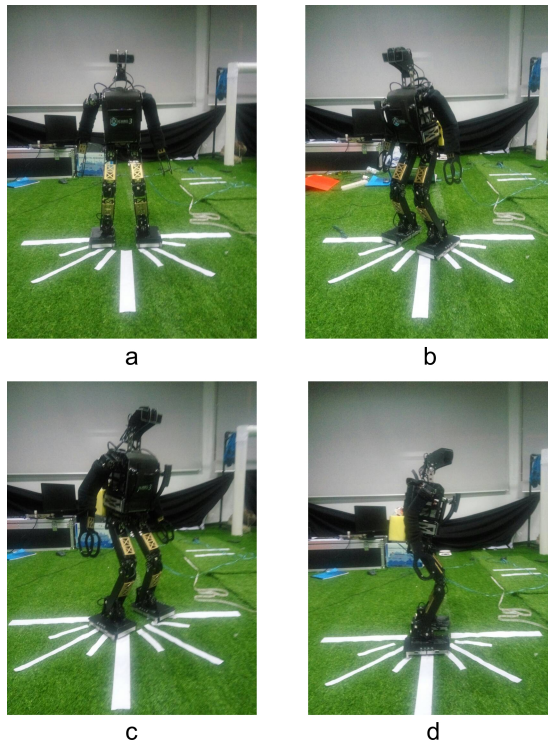


d

Gambar 4.1 Proses pengujian gerakan berjalan lurus ke depan

4.1.2 Gerakan Berputar

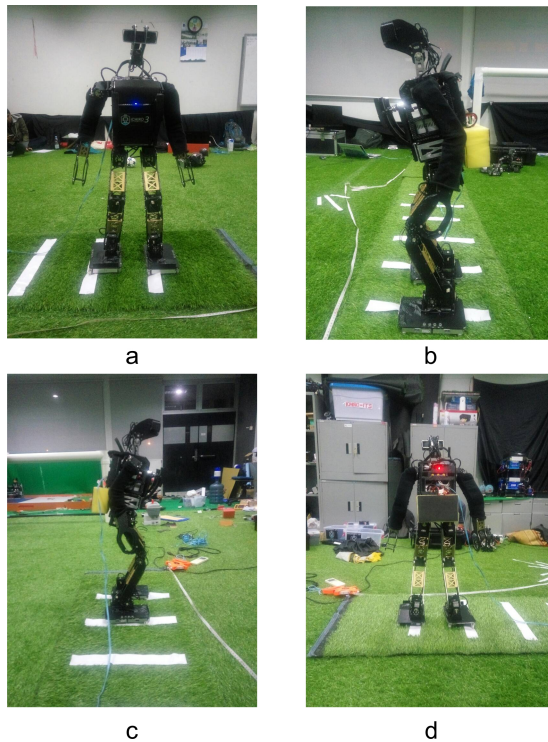
Pengujian gerakan berputar dilakukan dengan cara membuat garis melingkar pada sekitar robot seperti busur. Robot digerakkan berputar lalu diukur kecepatan angular berputarnya setiap kali melakukan pivot. Proses pengujian dapat dilihat pada gambar 4.2. Berdasarkan percobaan didapatkan data robot dapat berputar 30 derajat pada setiap kali melakukan gerakan pivot.



Gambar 4.2 Proses pengujian gerakan berjalan berputar

4.1.3 Gerakan Berjalan ke samping

Gerakan berjalan ke samping merupakan dasar dari pendekatan bola pada permainan sepak bola. Dengan adanya gerakan berjalan kesamping robot diharapkan dapat menjangkau bola yang ada di sisi depan-samping lebih cepat karena tidak harus melakukan gerakan berputar. Pengujian gerakan samping ini dilakukan dengan cara yang sama seperti pada gerakan berjalan lurus, hanya saja garis pengukur diletakkan disamping robot. Berdasarkan hasil pengujian didapatkan bahwa kecepatan berjalan robot adalah 7.5cm perdetik. Proses pengujian dapat dilihat pada gambar 4.3.



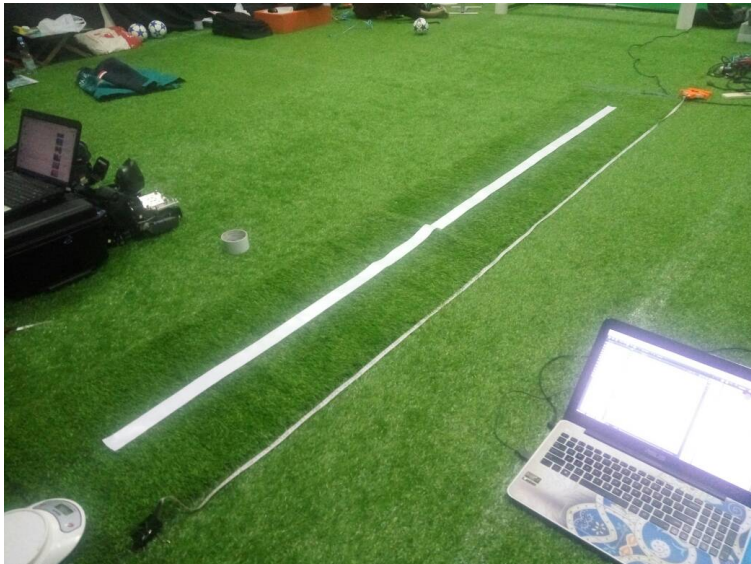
Gambar 4.3 Proses pengujian gerakan berjalan ke samping

4.2 Pengujian Keseimbangan Berjalan pada Akselerasi dan Deselerasi Robot

Pada pengujian ini dilakukan pengujian keseimbangan robot untuk membandingkan kondisi berjalan robot dalam kondisi normal dengan kondisi berjalan robot setelah ditambahkan akselerasi, kontrol keseimbangan pada hip, dan kontrol keseimbangan pada tangan. Hal ini bertujuan untuk memperoleh data semua kondisi berjalan robot sehingga bisa dianalisis apakah proses penambahan keseimbangan akselerasi dan deselerasi dapat berpengaruh pada performa robot.

Pengujian dilakukan dengan cara menjalankan robot sepanjang 3 meter dengan menggunakan garis putih sebagai lintasannya. Robot akan berhenti berjalan bila garis sudah hilang dari pandangan kamera robot. sudut kepala dibuat tetap, tidak bergerak mengangguk atau menoleh. Setiap pengujian robot juga diukur kecepatan robot untuk menempuh jarak lintasan yang disediakan. Lintasan pengujian dapat dilihat pada gambar 4.4.

Skenario pergerakan robot pada saat pengujian adalah robot dijalankan-



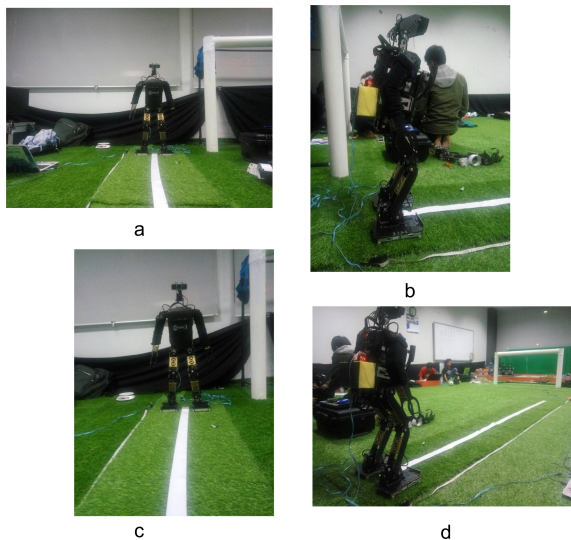
Gambar 4.4 Lintasan Pengujian Robot

an dengan kecepatan penuh pada awal geraknya untuk mengetahui proses akselerasi, lalu dipelankan pada iterasi ke 200 pada kecepatan setengah penuh untuk mengetahui efek perubahan kecepatan pada keseimbangan robot, lalu diberikan kecepatan penuh lagi pada iterasi ke 400 kemudian dihentikan pada iterasi ke 700 untuk mengetahui efek proses deselerasi.

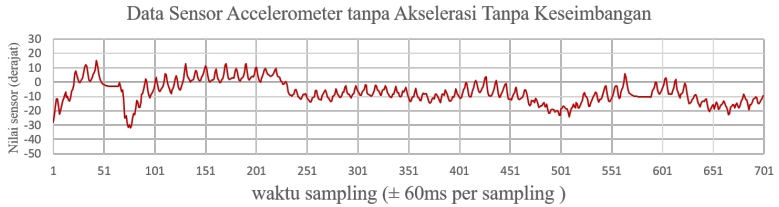
Data diakusisi menggunakan kontroler CM730, kemudian dikirim secara serial UART ke komputer utama NUC515RYK. Pengolahan data dilakukan di komputer utama. Data yang diterima komputer diolah diprint dengan format teks. Format teks tersebut dibuat di grafik menggunakan microsoft excel. Data yang divisualisasikan adalah data gyrometer dan accelerometer. Proses pengujian dapat dilihat pada gambar 4.5.

4.2.1 Data Hasil Pengujian Keseimbangan berjalan Robot tanpa Akselerasi, deselerasi, dan sistem keseimbangan

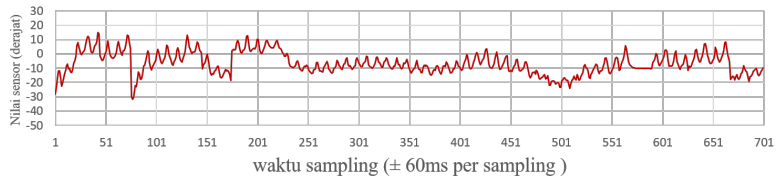
Pengujian ini dilakukan untuk mengetahui data sensor pada kondisi awal robot tanpa akselerasi dan keseimbangan tambahan. Dengan adanya data ini maka akan dapat dibandingkan seberapa besar efek optimasi yang



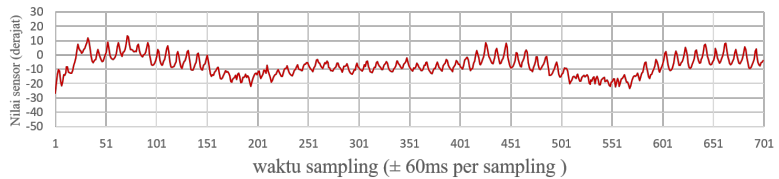
Gambar 4.5 pengujian Pengujian Robot



a. Data sensor accelerometer tanpa akselerasi tanpa keseimbangan percobaan 1



b. Data sensor accelerometer tanpa akselerasi tanpa keseimbangan percobaan 2



c. Data sensor accelerometer tanpa akselerasi tanpa keseimbangan percobaan 3

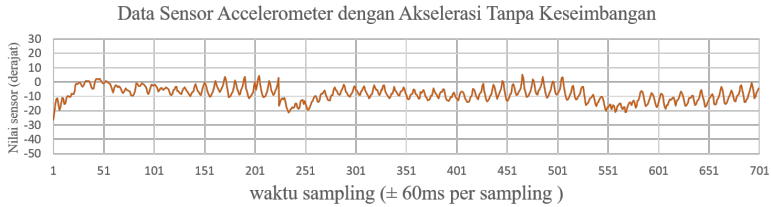
Gambar 4.6 Data sensor *accelerometer* Robot berjalan tanpa Akselerasi, deselerasi, dan sistem keseimbangan

dilakukan. grafik hasil pengujian dapat dilihat pada gambar 4.6

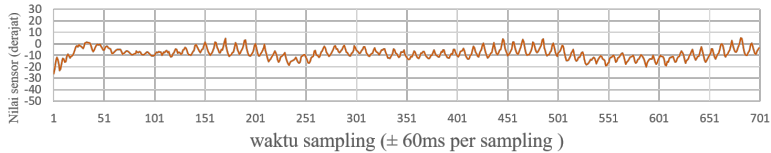
Berdasarkan data diatas dapat diketahui bahwa tanpa akselerasi, saat robot langsung di drve pada kecepatan 22 cm perdetik terdapat overshoot pada sensor *accelerometer*.

4.2.2 Data Hasil Pengujian Keseimbangan berjalan Robot dengan Akselerasi dan deselerasi tanpa sistem keseimbangan

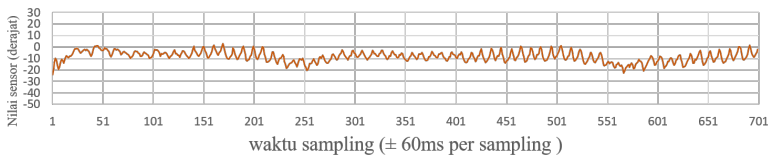
Dengan adanya akselerasi, robot diharapkan dapat melakukan gerakan berjalan lebih cepat pada lintasan yang disediakan. Pengujian ini dilakukan untuk mengetahui data sensor saat robot berjalan dengan akselerasi dan deselerasi tanpa adanya keseimbangan tambahan agar dapat diketahui ba-



a. Data sensor accelerometer dengan akselerasi tanpa keseimbangan percobaan 1



b. Data sensor accelerometer dengan akselerasi tanpa keseimbangan percobaan 2



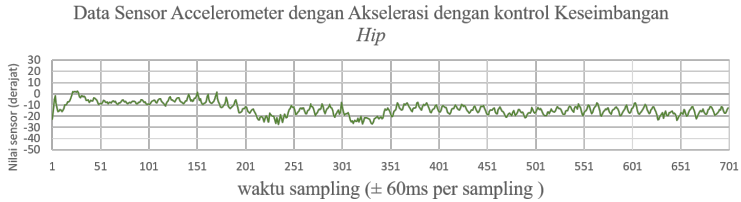
c. Data sensor accelerometer dengan akselerasi tanpa keseimbangan percobaan 3

Gambar 4.7 Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi tanpa sistem keseimbangan

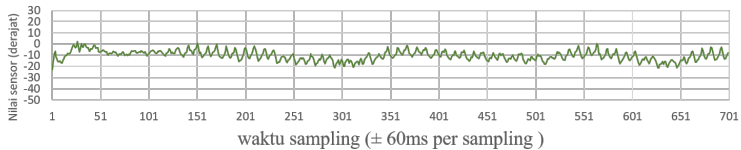
gaimana efek akselerasi dan deselerasi yang ditambahkan pada keseimbangan robot. Grafik dapat dilihat pada gambar 4.7

4.2.3 Data Hasil Pengujian Keseimbangan berjalan Robot dengan Akselerasi dan deselerasi dengan sistem keseimbangan *hip*

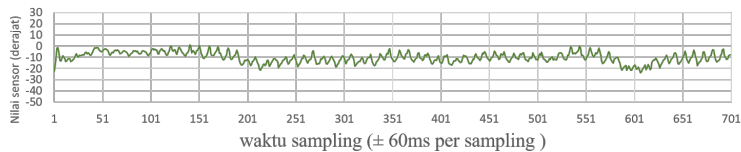
Pengujian ini dilakukan dengan mengaplikasikan sistem keseimbangan *hip* pada gerakan berjalan robot yang telah ditambahkan. Data yang didapatkan kemudian dibandingkan dengan pengujian lainnya apakah performa robot lebih baik atau lebih buruk terutama dengan pengujian sistem keseimbangan tangan. Grafik data sensor dapat dilihat pada gambar 4.8



a. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan
hip percobaan 1



b. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan
hip percobaan 2

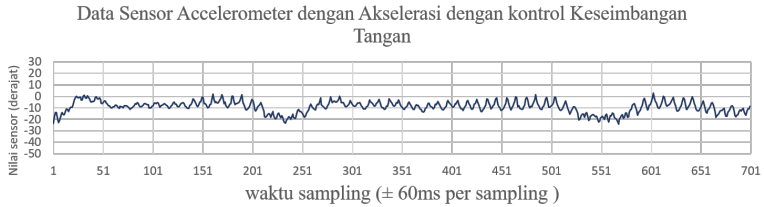


c. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan
hip percobaan 3

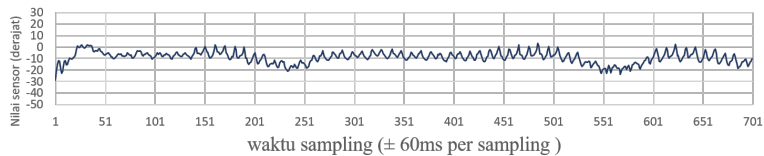
Gambar 4.8 Data sensor *accelerometer* Robot berjalan tanpa Akselerasi dan de-selerasi dengan sistem keseimbangan *hip*

4.2.4 Data Hasil Pengujian Keseimbangan berjalan Robot dengan Akselerasi, deselerasi, dan sistem keseimbangan tangan

Pengujian ini dilakukan dengan mengaplikasikan sistem keseimbangan tangan pada gerakan berjalan robot yang telah ditambahkan. Data yang didapatkan kemudian dibandingkan dengan pengujian lainnya apakah performa robot lebih baik atau lebih buruk terutama dengan pengujian sistem keseimbangan *hip*. Grafik data sensor dapat dilihat pada gambar 4.9



b. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan tangan percobaan 2

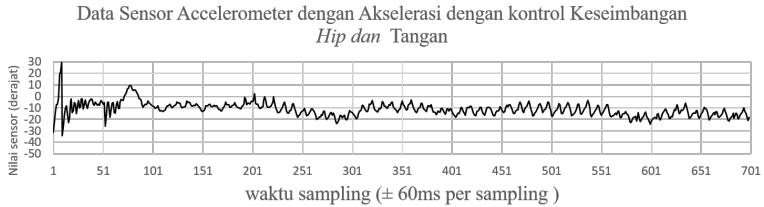


c. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan tangan percobaan 3

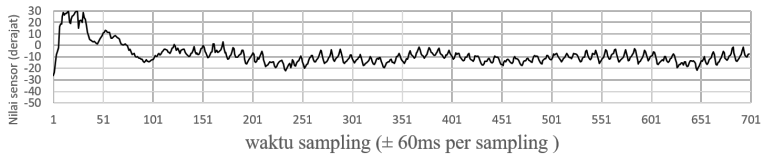
Gambar 4.9 Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi dengan sistem keseimbangan tangan

4.2.5 Data Hasil Pengujian Keseimbangan berjalan Robot dengan Akselerasi, deselerasi, dan sistem keseimbangan *hip* dan tangan

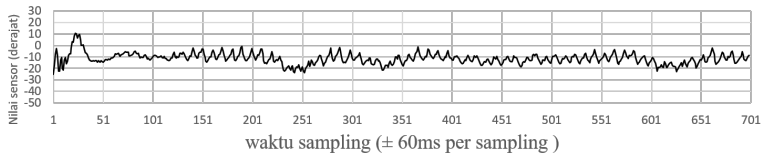
Pengujian ini dilakukan dengan mengaplikasikan sistem keseimbangan tangan dan *hip* pada gerakan berjalan robot yang telah ditambahkan. Data yang didapatkan digunakan untuk mengetahui bagaimana jika sistem keseimbangan *hip* dan tangan digabungkan. Grafik data sensor dapat dilihat pada gambar 4.10



a. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan
hip dan tangan percobaan 1



b. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan
hip dan tangan percobaan 2



c. Data sensor accelerometer dengan akselerasi dengan kontrol keseimbangan
hip dan tangan percobaan 3

Gambar 4.10 Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi dengan sistem keseimbangan *hip* dan tangan

4.3 Analisa Hasil Pengujian Sistem Keseimbangan Berjalan Robot

Subbab ini akan menjelaskan perbandingan hasil pengujian sistem keseimbangan robot yang telah diuji. Proses pengujian yang telah dilakukan adalah dengan menjalankan robot dengan kecepatan penuh dari keadaan diam, lalu dipelankan beberapa saat, setelah itu dipercepat lagi menjadi kecepatan penuh, dan yang terakhir adalah robot dihentikan dari kecepatan penuh.

Hal yang dibandingkan pada subbab ini adalah kondisi awal saat robot langsung diberi kecepatan penuh, kondisi saat robot dipelankan jalannya,

dan kondisi saat robot dihentikan dari kecepatan penuh.

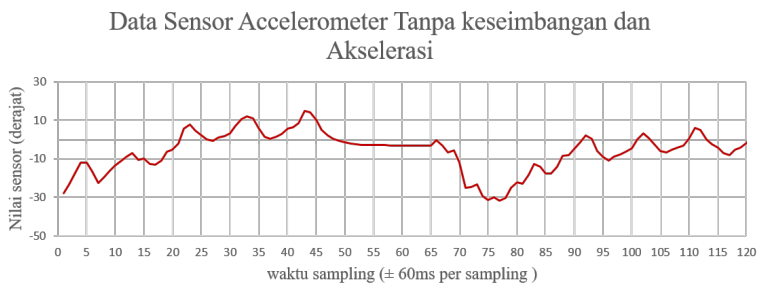
4.3.1 Perbandingan Kondisi berjalan Robot Saat Mulai Berjalan dari Keadaan Berhenti

Berikut adalah perbandingan data setiap pengujian. Data pada gambar 4.11 menunjukkan bahwa nilai nilai sensor berosilasi dan cenderung naik hingga pada sampling ke 50. Berdasarkan pengamatan visual pada robot saat pengujian robot terjatuh sebelum sampling ke 50 sehingga harus dipegang untuk melanjutkan pengujian

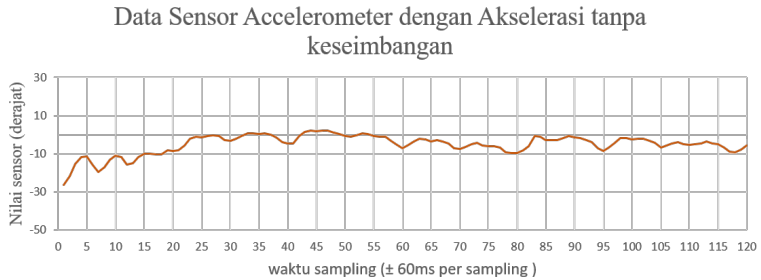
Gambar 4.11 menunjukkan bahwa jika diberi sistem akselerasi, robot akan cenderung lebih lambat saat mulai jalan karena harus jalan ditempat dulu. Pada pngujian ini robot tidak jatuh saat berjalan sehingga robot dapat berjalan dengan kecepatan tinggi tanpa terjatuh. Namun robot cenderung tidak stabil saat berjalan tanpa adanya kontrol.

Gambar 4.13 menunjukkan bahwa saat robot diberi kontrol keseimbangan pada *hip* proses menuju *steady state* menjadi lebih cepat. Sebelum iterasi 45 sistem sudah *steady state* sehingga dapat dikatakan bahwa sistem sudah bekerja dengan baik jika diberi kontrol keseimbangan pada *hip*.

Pada saat pengujian robot dengan kontrol keseimbangan lengan, proses menuju *steady state* terjadi lebih lambat dibandingkan saat robot diberi kontrol keseimbangan pada *hip* namaun masih lebih cepat dibandingkan jika robot tidak diberi kontrol apapun. grafik data hasil pengujian robot



Gambar 4.11 Data sensor *accelerometer* Robot berjalan tanpa Akselerasi, deselerasi, dan sistem keseimbangan

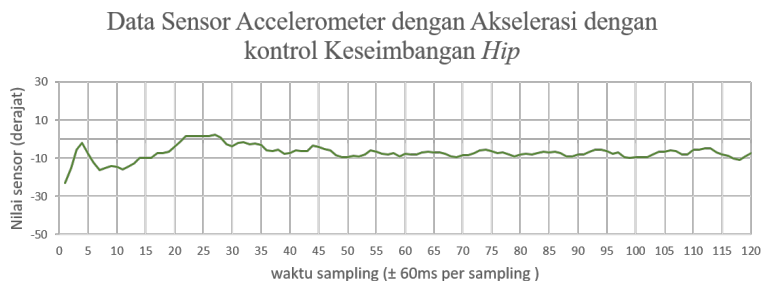


Gambar 4.12 Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi, tanpa sistem keseimbangan

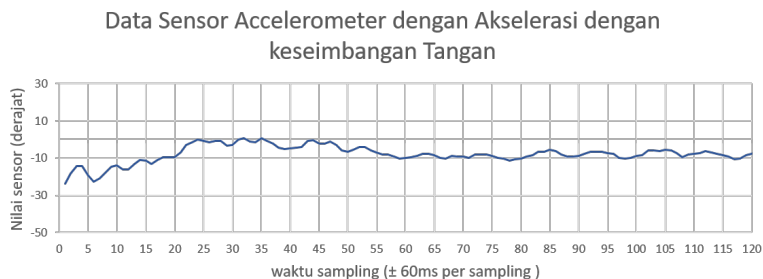
dengan kontrol keseimbangan lengan dapat dilihat pada gambar 4.14.

Jika kontrol keseimbangan pada *hip* dan tangan diterapkan bersama-sama tanpa mengganti nilai PID pada masing-masing kontrol, terjadi overshoot yang berlebihan sehingga nilai *steady state*-nya juga lebih lama melebihi iterasi ke 50 sistem baru bisa *steady state*

Berdasarkan uraian diatas, dapat diketahui bahwa terdapat perbedaan yang signifikan antara pengujian berjalan robot tanpa akselerasi dan pengujian berjalan robot dengan akselerasi. Tanpa adanya akselerasi, jika robot langsung dijalankan dengan kecepatan tinggi maka terjadi overshoot pada nilai sensor. Berdasarkan 3 kali percobaan pengujian ketiganya



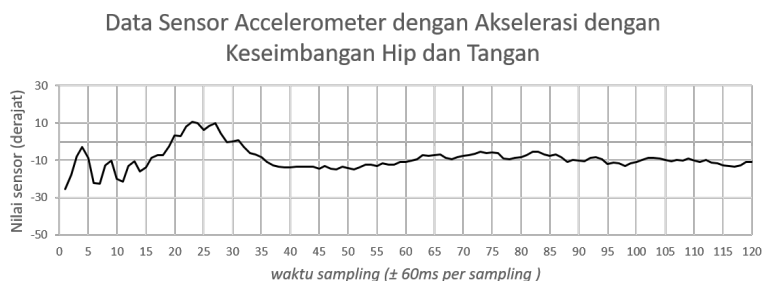
Gambar 4.13 Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi, dengan kontrol keseimbangan *hip*



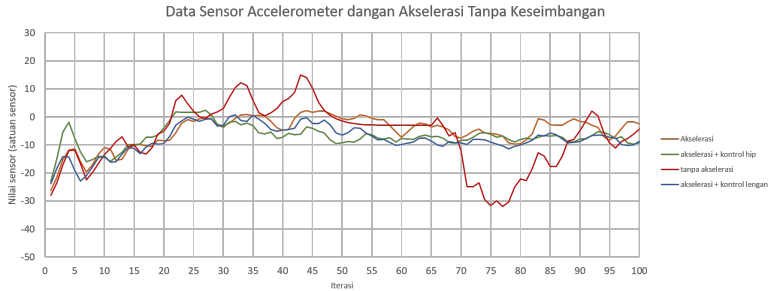
Gambar 4.14 Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi, dengan kontrol keseimbangan tangan

jatuh pada saat langkah pertama. Sedangkan jika diberi percepatan dari 3 kali percobaan robot tidak jatuh tetapi cenderung jatuh karena tidak diseimbangkan.

Pada saat sistem diberi keseimbangan *hip*, proses menuju steady state menjadi lebih cepat dan lebih stabil. Seperti halnya dengan keseimbangan tangan, sistem lebih stabil pada saat kondisi awal berjalan. Namun jika sistem keseimbangan digabungkan tanpa mengubah nilai PID, terjadi overshoot berlebihan pada robot karena nilai proporsional kontrol yang saling menjumlahkan. Perbandingan Hasil Pengujian saat robot berakselerasi dapat dilihat pada gambar 4.16. Untuk grafik yang lebih jelas dapat dilihat



Gambar 4.15 Data sensor *accelerometer* Robot berjalan dengan Akselerasi dan deselerasi, dengan kontrol keseimbangan *hip* dan tangan



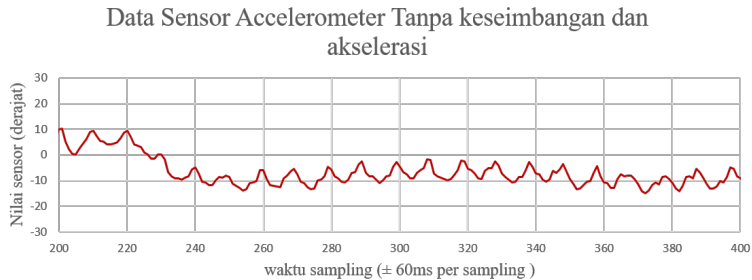
Gambar 4.16 Perbandingan data sensor *accelerometer* pengujian Robot berjalan saat akselerasi .

di lampiran.

Grafik pada gambar 4.16 menunjukkan data sensor pada pengujian. Kondisi steady state diatur dengan set point sensor pada sudut -10 sampai -5 derajat. Berdasarkan data tersebut diketahui bahwa keseimbangan berjalan robot tanpa akselerasi dan kontrol keseimbangan pada grafik warna merah sangat tidak stabil, bahkan robot sempat jatuh pada sampling ke 30 (sekitar 2 detik). Sementara ketika diberikan proses percepatan (grafik orange), sistem dapat mencapai kestabilan pada sampling ke 67 (sekitar 4 detik). Pada saat sistem diberi proses akselerasi dan kontrol keseimbangan pada bagian hip (grafik hijau), robot dapat mencapai kondisi kestabilan berjalan pada sampling ke 45 (sekitar 2,5 detik). Terakhir, jika sistem diberi kontrol keseimbangan tangan sistem mencapai steady state pada sampling ke 53 (sekitar 3 detik). Berdasarkan perbandingan data tersebut didapatkan sistem yang paling baik adalah pada kontrol keseimbangan pada bagian hip. Hal ini disebabkan karena kontrol keseimbangan pada bagian hip berperan langsung pada proses penstabilan.

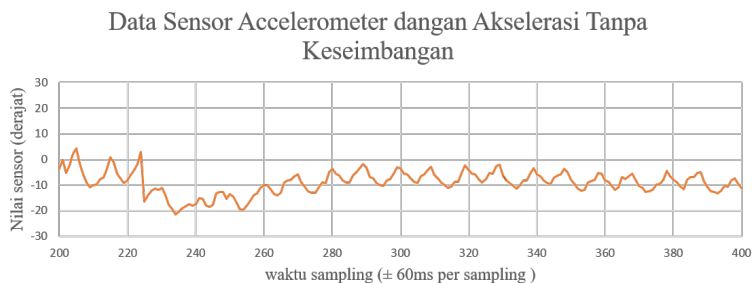
4.3.2 Perbandingan Kondisi berjalan robot saat diperlambat

Bagian ini akan menjelaskan perbandingan hasil pengujian pada saat jalan robot diperlambat. Proses pengurangan nilai kecepatan dimulai pada iterasi ke 200 lalu dipercepat lagi menjadi kecepatan maksimum pada iterasi ke 400. Kecepatan robot diturunkan dari nilai maksimum 23 ke 10 (22 cm per detik menjadi 10 cm perdetik).

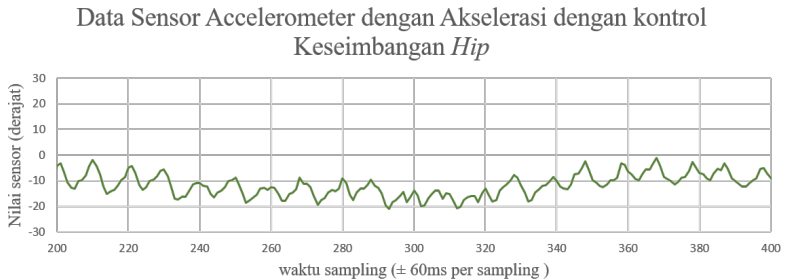


Gambar 4.17 Data sensor *accelerometer* Robot berjalan diperlambat tanpa Akselerasi, deselerasi, dan kontrol keseimbangan

Gambar 4.17 menunjukkan nilai sensor pada saat robot dierlambat dari dengan kondisi tanpa akselerasi dan sistem keseimbangan. Dapat dilihat bahwa robot menjadi tidak stabil dan cenderung terhuyung saat diperlambat. pada saat dipercepat lagi robot cenderung terhuyung kearah yang sama dengan saat diperlambat, seharusnya arahnya berbeda. Hal ini disebabkan karena badan mengikuti osilasi dan kebetulan saat dipercepat badan terosilasi ke arah yang sama dengan proses perlambatan sehingga saat dipercepat robot cenderung terhuyung ke arah depan juga. Dengan ditambahkan perlambatan lalu dipercepat kembali, robot mengalami osilasi berlebihan. Kondisi ini sangat rawan untuk bermain pertandingan sepak



Gambar 4.18 Data sensor *accelerometer* Robot berjalan diperlambat dengan Akselerasi dan deselerasi, tanpa kontrol keseimbangan

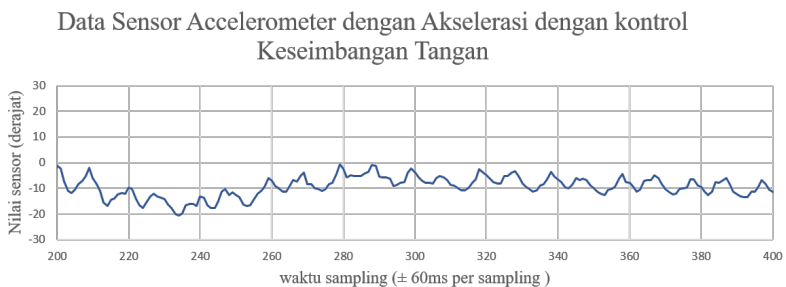


Gambar 4.19 Data sensor *accelerometer* Robot berjalan diperlambat dengan Akselerasi dan deselerasi, dan kontrol keseimbangan *hip*

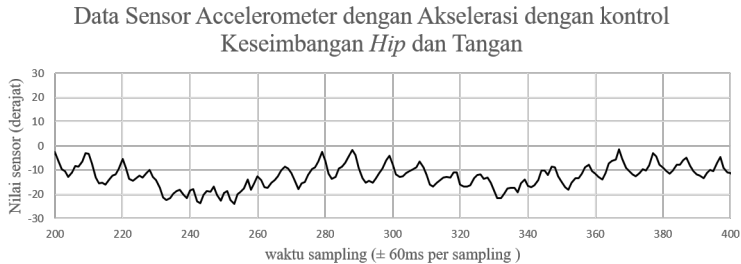
bola karena terdapat faktor eksternal seperti bersenggolan dengan lawan sehingga robot akan lebih mudah terjatuh.

Pada saat ditambahkan sistem akselerasi, terlihat pada gambar 4.18 bahwa sistem sudah cukup stabil. Hal ini menunjukkan bahwa dengan menambahkan proses akselerasi dan deselerasi saja sudah dapat mengatasi osilasi pada proses perlambatan berjalan robot.

Terlihat pada gambar 4.19 bahwa sistem juga sudah cukup stabil. Pada pengujian tersebut ditambahkan sistem akselerasi dan kontrol keseimbangan *hip*. Namun sistem sempat mempertahankan nilai dibawah set poin mulai iterasi 200 hingga 350, hal ini terjadi karena pada proses kon-



Gambar 4.20 Data sensor *accelerometer* Robot berjalan berjalan diperlambat dengan Akselerasi dan deselerasi, dan kontrol keseimbangan tangan

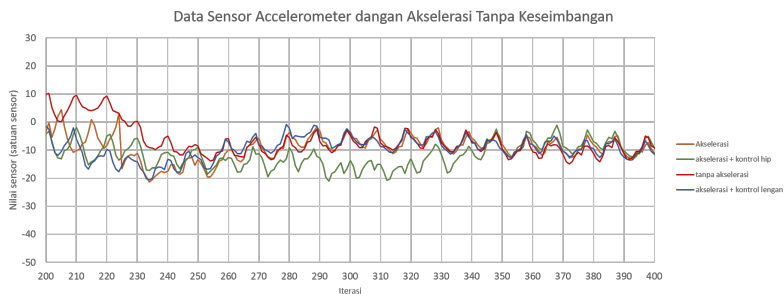


Gambar 4.21 Data sensor *accelerometer* Robot berjalan tanpa Akselerasi, deselerasi, dan kontrol keseimbangan

trol PID diberi nilai *offset* pada set poin sebesar 5 derajat untuk mencegah osilasi.

Pada saat ditambahkan sistem akselerasi dan kontrol keseimbangan tangan, terlihat pada gambar 4.20 bahwa sistem juga sudah cukup stabil. Sistem sempat mengalami sedikit osilasi karena terhuyung ke depan sehingga proses penyesuaian baru bisa stabil pada iterasi ke 315.

Sedangkan pada saat kontrol keseimbangan hip dan tangan diterapkan bersama, sistem berosilasi berlebihan pada proses perlambatan, sistem baru bisa stabil pada iterasi ke 400. hal ini juga disebabkan karena nilai Proporsional yang terlalu berlebihan.



Gambar 4.22 Perbandingan data sensor *accelerometer* pengujian Robot saat berjalan dengan deselerasi .

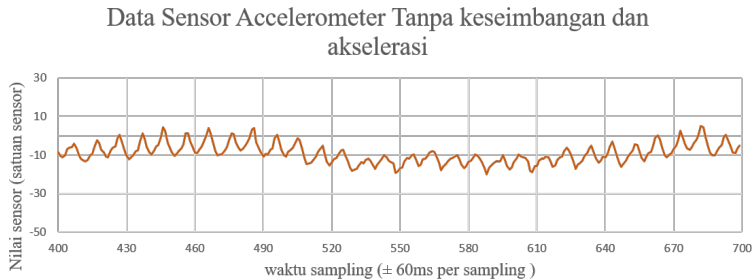
Berdasarkan uraian diatas dapat disimpulkan bahwa sistem yang paling bagus saat deselerasi adalah pada sistem dengan kontrol keseimbangan *hip* karena tidak berosilasi. Namun pada percobaan yang lain, terdapat satu kali percobaan dimana sistem kontrol keseimbangan *hip* justru berosilasi secara berlebihan. Perbandingan Hasil Pengujian saat robot berakselerasi dapat dilihat pada gambar 4.22. Untuk grafik yang lebih jelas dapat dilihat di lampiran

Grafik pada gambar 4.22 menunjukkan data keseimbangan robot ketika robot diperlambat. . Kondisi steady state diatur dengan set point sensor pada sudut antara -10 sampai -5 derajat. Proses deselerasi dimulai dari iterasi ke 200 (5 detik) Kondisi keseimbangan berjalan robot tanpa akselerasi dan kontrol keseimbangan dapat dilihat pada grafik warna merah. Dapat dilihat bahwa sistem dapat mencapai steady state pada nilai sampling ke 315 (sekitar 10 detik). Sementara ketika diberikan proses percepatan (grafik orange), sistem dapat mencapai kestabilan pada sampling ke 296 (sekitar 9 detik). Pada saat sistem diberi proses akselerasi dan kontrol keseimbangan pada bagian hip (grafik hijau), robot dapat mencapai kondisi kestabilan berjalan pada sampling ke 350 (sekitar 12 detik). Terakhir, jika sistem diberi kontrol keseimbangan tangan sistem mencapai steady state pada sampling ke 275 (sekitar 8 detik). Berdasarkan perbandingan data tersebut didapatkan sistem yang paling baik adalah pada kontrol keseimbangan pada bagian tangan.

4.3.3 Perbandingan Kondisi berjalan robot saat dihentikan

roses ini sangat penting karena berhentinya robot pada pertandingan sepak bola sangat sering terjadi dan sangat rawan bersenggolan dengan musuh, sehingga dibutuhkan sistem yang benar-benar stabil dalam hal ini. Bagian ini akan menjelaskan perbandingan hasil pengujian pada saat jalan robot berhenti. Proses pergantian nilai kecepatan menjadi 0 dimulai pada iterasi ke 600 dan penghentian pencatatan nilai sensor pada iterasi ke 700.

Berdasarkan data sensor pada gambar 4.23 Dapat dilihat bahwa sejak sebelum proses berhenti sistem sudah mengalami osilasi secara berlebihan. Bahkan, diantara iterasi 550 dan 600 robot sempat jatuh dan harus dipegang untuk melanjutkan pengujian. Pada saat deselerasi menuju kecepatan nol pun robot tetap osilasi. Hal ini dapat menyebabkan robot mudah terjatuh jika didepannya ada halangan atau bola. Dengan ini dapat disimpulkan bahwa proses berhenti robot tanpa deselerasi sangat tidak

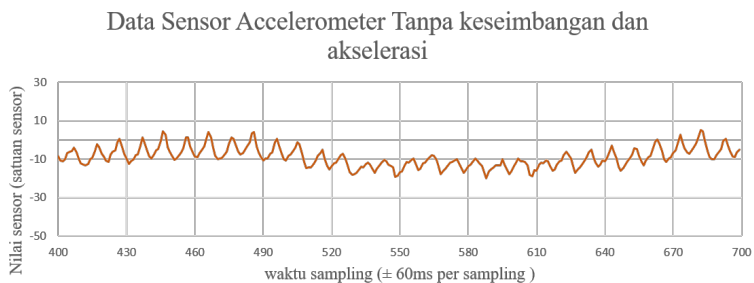


Gambar 4.23 Data sensor *accelerometer* Robot dihentikan tanpa Akselerasi, deselerasi, dan kontrol keseimbangan

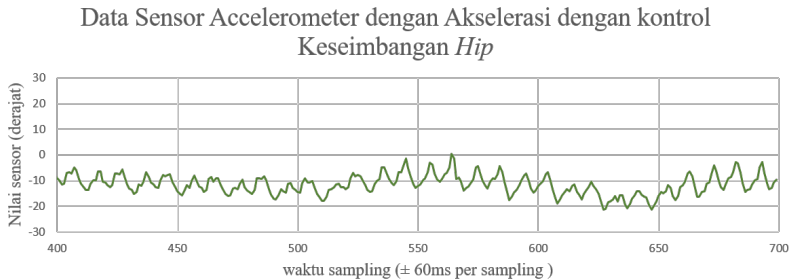
dianjurkan karena robot dapat jatuh.

Pada pengujian sistem ketika diberi proses akselerasi, dapat dilihat pada gambar 4.24 bahwa sistem sudah stabil. Hal ini menunjukkan bahwa dengan menambahkan proses akselerasi dan deselerasi juga sudah dapat mengatasi osilasi pada proses perlambatan berjalan robot dari kecepatan maksimum 22 cm perdetik menjadi berhenti

Berdasarkan data sensor pada gambar 4.25, dapat disimpulkan bahwa robot juga sudah dapat bekerja dengan baik. Hal ini menunjukkan bahwa dengan menambahkan proses akselerasi dan deselerasi dan kontrol keseimbangan hip juga sudah dapat mengatasi osilasi pada proses perlambatan



Gambar 4.24 Data sensor *accelerometer* Robot dihentikan dengan Akselerasi dan deselerasi tanpa kontrol keseimbangan



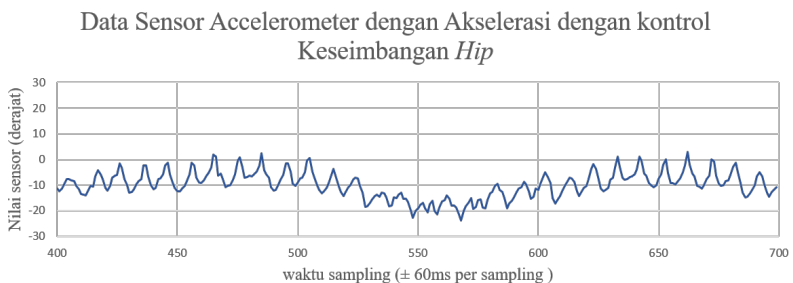
Gambar 4.25 Data sensor *accelerometer* Robot dihentikan dengan Akselerasi dan deselerasi dengan kontrol keseimbangan *hip*

berjalan robot. Namun pada saat akselerasi kembali ke kecepatan maksimum, sistem sempat berosilasi.

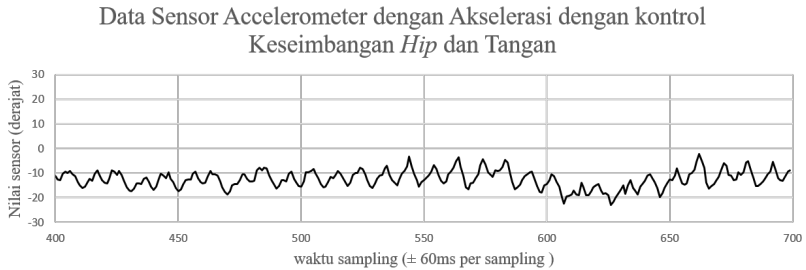
Pada pengujian sistem ketika diberi proses akselerasi dan kontrol keseimbangan tangan, dapat dilihat pada gambar 4.26 bahwa sistem sudah stabil. Namun pada saat robot berjalan pada kecepatan bernilai 10, sistem sempat mengalami osilasi pada iterasi ke 500 sampai 600.

Pada saat kontrol keseimbangan hip dan tangan diterapkan bersama, sistem mengalami osilasi mulai iterasi ke 550 dan belum dapat mencapai steady state hingga pengambilan data nilai sensor berakhir.

Berdasarkan uraian diatas, sama seperti pada proses perlambatan, da-

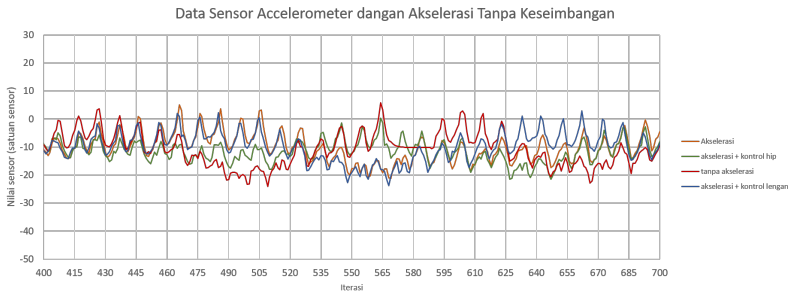


Gambar 4.26 Data sensor *accelerometer* Robot dihentikan dengan Akselerasi dan deselerasi dengan kontrol keseimbangan tangan



Gambar 4.27 Data sensor *accelerometer* Robot dihentikan dengan Akselerasi dan deselerasi dengan kontrol keseimbangan *hip* dan tangan

pat disimpulkan bahwa sistem yang paling bagus untuk proses berhenti robot adalah pada sistem dengan kontrol keseimbangan *hip* karena tidak berosilasi. Perbandingan Hasil Pengujian saat robot berhenti dapat dilihat pada gambar 4.28. Untuk grafik yang lebih jelas dapat dilihat di lampiran



Gambar 4.28 Perbandingan data sensor *accelerometer* pengujian Robot berjalan saat akan berhenti.

Grafik pada gambar 4.28 menunjukkan data keseimbangan robot ketika robot dihentikan. Kondisi steady state diatur dengan set point sensor pada sudut antara -10 sampai -5 derajat. Kondisi keseimbangan berjalan robot tanpa akselerasi dan kontrol keseimbangan dapat dilihat pada grafik warna merah. Dapat dilihat bahwa sistem tidak dapat mencapai steady state. Sementara ketika diberikan proses percepatan (grafik orange), sis-

tem dapat mencapai kestabilan pada sampling ke 654 (sekitar 16 detik). Pada saat sistem diberi proses akselerasi dan kontrol keseimbangan pada bagian hip (grafik hijau), robot dapat mencapai kondisi kestabilan berjalan pada sampling ke 615 (sekitar 14 detik). Terakhir, jika sistem diberi kontrol keseimbangan tangan sistem mencapai steady state pada sampling ke 411 (sekitar 16 detik). Berdasarkan perbandingan data tersebut didapatkan sistem yang paling baik adalah pada kontrol keseimbangan pada bagian hip.

Halaman ini sengaja dikosongkan.

BAB 5

PENUTUP

5.1 Kesimpulan

Pada tugas akhir ini telah berhasil dilakukan Kontrol Keseimbangan *Upper Body* pada Proses Akselerasi dan Deselerasi Gerakan Berjalan Robot Sepak Bola Humanoid. Dari tugas akhir ini, didapatkan kesimpulan sebagai berikut:

1. Desain mekanik robot Ithaaro berhasil diimplementasikan, robot dapat bergerak dengan kecepatan 23 cm perdetik pada jalan lurus, 7.5 cm perdetik pada jalan kesamping, dan berputar 30 derajat pada gerakan berputar setiap langkahnya.
2. Proses Akselerasi dan deselerasi dapat menambah kesabilan berjalan sehingga robot tidak terjatuh saat bergerak dari diam ke berjalan dengan kecepatan 0,23m/s.
3. Dengan menerapkan kontrol PD metode *metode trial and eror* dengan nilai P sebesar 0.04, dan D sebesar 0.02 pada aktuator robot bagian *hip pitch*, gerakan berjalan robot menjadi lebih stabil dengan mencapai steady state dalam 2.5 detik.

5.2 Saran

Untuk pengembangan lebih lanjut mengenai tugas akhir ini, disarankan untuk melakukan beberapa langkah lanjutan:

1. Penyempurnaan desain mekanik dengan analisis lebih dalam pada proses desain agar performa robot lebih maksimal.
2. Menggunakan metode lain dalam proses akselerasi dan deselerasi.
3. mencoba permodelan lain pada keseimbangan saat berjalan dengan akselerasi dan deselerasi.

Halaman ini sengaja dikosongkan.

DAFTAR PUSTAKA

- [1] RoboCup Soccer Humanoid League Rules and Setup. (Dikutip pada halaman 7).
- [2] Peraturan dan Persiapan Kontes Robot Sepak Bola Indonesia. No citations.
- [3] I. Ha, Y. Tamura, and H. Asama, "Development of open platform hu- manoid robot DARwIn-OP," Advanced Robotics, vol. 27, pp. 223– 232, Feb. 2013. (Dikutip pada halaman 2180).
- [4] Supriyanto Raden. . 2010. "Buku Ajar Robotika," Jakarta. Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Gunadarma. (Dikutip pada halaman 40 dan 47-52).
- [5] Tözeren Aydın. 2000 "Human Body Dynamics: Classical Mechanics and Human Movement". Washington. Department of Biomedical Engineering The Catholic University of America. No Citation
- [6] Schwarz, Max et all. "NimbRo-OP humanoid teensize open platform." In In Proceedings of 7th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots, Osaka. 2012. Dikutip pada halaman2-3.
- [7] Suryawan Donny, Apriyanto Feri, Nurgaesang Faisal Arif. 2013 "Implementasi *Dynamic Walking* pada *Humanoid Robot Soccer*" Universitas Islam Indonesia. No citations.
- [8] Astrom K, Hagglund T. 1995. "PID Controllers: Theory, Design, and Tunning. 2nd Edition." Instrument Society of America. No citations
- [9] <https://blog.lxde.org/2009/02/05/lubuntu-lxde-meet-up-with-mark-shuttleworth-in-berlin/>. Diakses pada tanggal 12 Juli 2017 pukul 11.55.

Halaman ini sengaja dikosongkan.

LAMPIRAN

1. Program Akselerasi

```
const double FB_ACC_ALPHA = 0.2;
const double FB_DEC_ALPHA = 0.3;

if(x_input > m_X_Move_Amplitude)
    m_X_Move_Amplitude = (1.0-FB_ACC_ALPHA) *
m_X_Move_Amplitude + FB_ACC_ALPHA * x_input;
else if(x_input < m_X_Move_Amplitude)
    m_X_Move_Amplitude = (1.0-FB_DEC_ALPHA) *
m_X_Move_Amplitude + FB_DEC_ALPHA * x_input;
```

dengan keterangan :

FB_ACC_ALPHA = faktor pengali nilai akselerasi

FB_DEC_ALPHA = faktor pengali nilai deselerasi

m_X_Move_Amplitude = hasil output percepatan

x_input = buffer untuk data kecepatan yang diinputkan

2. Program Kontrol PID Hip

```
#define m_x_p_gain_hip 0.04
#define m_x_i_gain_hip 0
#define m_x_d_gain_hip 0.02

error_x_hip = COG - fbAccelErr;
iOffset_x_hip = m_x_i_gain_hip*error_x_hip;
m_x_sum_int_hip = m_x_sum_int_hip + iOffset_x_hip;
dOffset_x_hip = m_x_d_gain_hip*error_x_hip;
m_x_err_diff_hip      =      dOffset_x_hip      -
m_x_err_diff_old_hip;
m_x_err_diff_old_hip = dOffset_x_hip;
```



```

out_pid_x_hip =
m_x_p_gain_hip*(error_x_hip+m_x_sum_int_hip+m_x_err_diff_hip);
//~ out_pid_x_hip_1 = out_pid_x_hip_1 + out_pid_x_hip;

if(fabs(error_x) < 20) out_pid_x_hip_1 = out_pid_x_hip_1;
else out_pid_x_hip_1 = out_pid_x_hip_1 + out_pid_x_hip;

```

dengan keterangan :

m_x_p_gain_hip = nilai KProporsional
 m_x_i_gain_hip = nilai KIntegral
 m_x_d_gain_hip = nilai KDiferensial
 fbAccelErr = nilai sensor
 COG = set poin nilai COG
 error_x_hip = nilai *error* set poin
 m_x_sum_int_hip = $K_i * e$
 m_x_err_diff_hip = $K_d * e$
 iOffset_x_hip = $K_i * error$
 dOffset_x_hip = $K_i * error$
 out_pid_x_hip = output PID

3. Program Kontrol PID tangan

```

#define m_x_p_gain 30
#define m_x_i_gain 0
#define m_x_d_gain 3

error_x = COG - fbAccelErr;
iOffset_x = m_x_i_gain*error_x;
m_x_sum_int = m_x_sum_int + iOffset_x;
dOffset_x = m_x_d_gain*error_x;
m_x_err_diff = dOffset_x - m_x_err_diff_old;
m_x_err_diff_old = dOffset_x;

out_pid_x =

```

```

m_x_p_gain*(error_x+m_x_sum_int+m_x_err_diff);
if(fabs(error_x) <20) errorTangan = 1;
else errorTangan = out_pid_x;

```

dengan keterangan :

```

m_x_p_gain = nilai KProporsional
m_x_i_gain = nilai KIntegral
m_x_d_gain = nilai KDiferensial
fbAccelErr = nilai sensor
COG = set poin nilai COG
error_x = nilai error set poin
m_x_sum_int =  $K_i * e$ 
m_x_err_diff =  $K_d * e$ 
iOffset_x =  $K_i * error$ 
dOffset_x =  $K_i * error$ 
out_pid_x = output PID

```

4. Program Pengujian Akselerasi dan deselerasi

```

case 0:
{

    data_accelero_pitch = MotionStatus::FB_ACCEL;
    data_gyro_pitch = MotionStatus::FB_GYRO;

    cout << "-----Ready Mode-----" << endl;

    Walking::GetInstance()-
    >m_Joint.SetEnableBodyWithoutHead(true, true);
    hitung++;
    if (hitung>50 && hitung <300)
    {
        Walking::GetInstance()-
        >X_MOVE_AMPLITUDE= 23;
        Walking::GetInstance()->Start();
        if (myfile.is_open())
        {
            myfile << "b";

```

```

        myfile << "                ";
        myfile << MotionStatus::ANGLE_PITCH;
        myfile << "                ";
        myfile << data_gyro_pitch;
        myfile << "\n";
    }
    else cout << "Unable to open file";
}
else if (hitung>300 && hitung <500)
{
    Walking::GetInstance()-
>X_MOVE_AMPLITUDE= 10;
    Walking::GetInstance()->Start();
    if (myfile.is_open())
    {
        myfile << "c";
        myfile << "                ";
        //~ myfile << data_accelero_pitch;
        myfile << MotionStatus::ANGLE_PITCH;
        myfile << "                ";
        myfile << data_gyro_pitch;
        myfile << "\n";
    }
    else cout << "Unable to open file";
}
else if (hitung>500 && hitung <800)
{
    Walking::GetInstance()-
>X_MOVE_AMPLITUDE= 20;
    Walking::GetInstance()->Start();
    if (myfile.is_open())
    {
        myfile << "c";
        myfile << "                ";
        //~ myfile << data_accelero_pitch;
        myfile << MotionStatus::ANGLE_PITCH;
        myfile << "                ";
        myfile << data_gyro_pitch;
        myfile << "\n";
    }
}

```

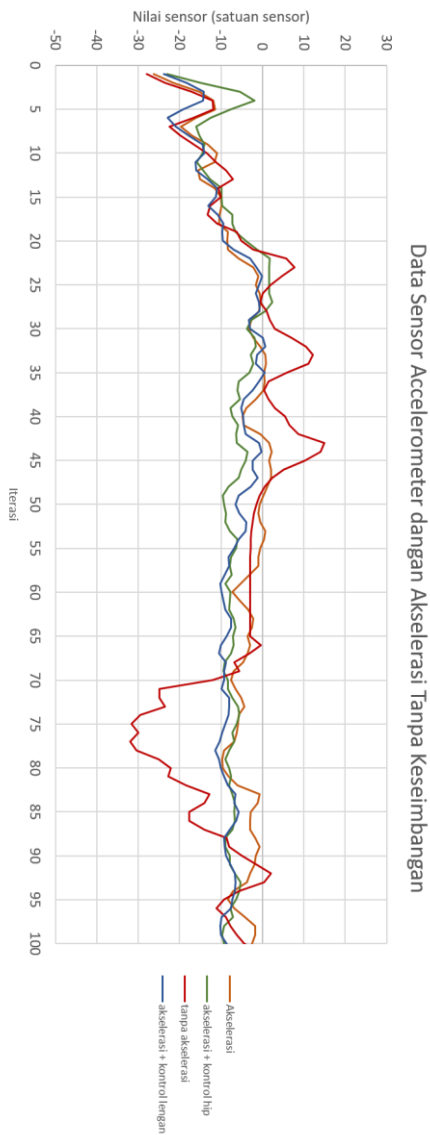
```

        else cout << "Unable to open file";
    }
    else if (hitung>800)
    {
        Walking::GetInstance()->Stop();
        myfile.close();
    }
    else
    {
        Walking::GetInstance()-
>X_MOVE_AMPLITUDE = 0;
        Walking::GetInstance()->Start();
        if (myfile.is_open())
        {
            myfile << "a";
            myfile << "          ";
            //~ myfile << data_accelero_pitch;
            myfile << MotionStatus::ANGLE_PITCH;
            myfile << "          ";
            myfile << data_gyro_pitch;
            myfile << "\n";
        }
        else cout << "Unable to open file";

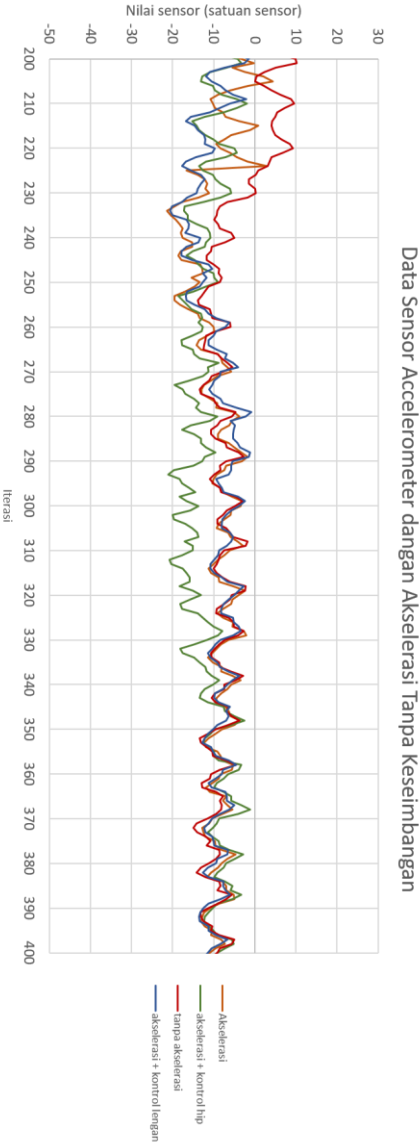
        cout << "jalan: "<<Walking::GetInstance()-
>X_MOVE_AMPLITUDE << endl;
        cout << "pitch accelerometer:" <<
MotionStatus::ANGLE_PITCH<< endl;
        cout << "pitch gyro:" << data_gyro_pitch<< endl;
    }
    break;
}

```

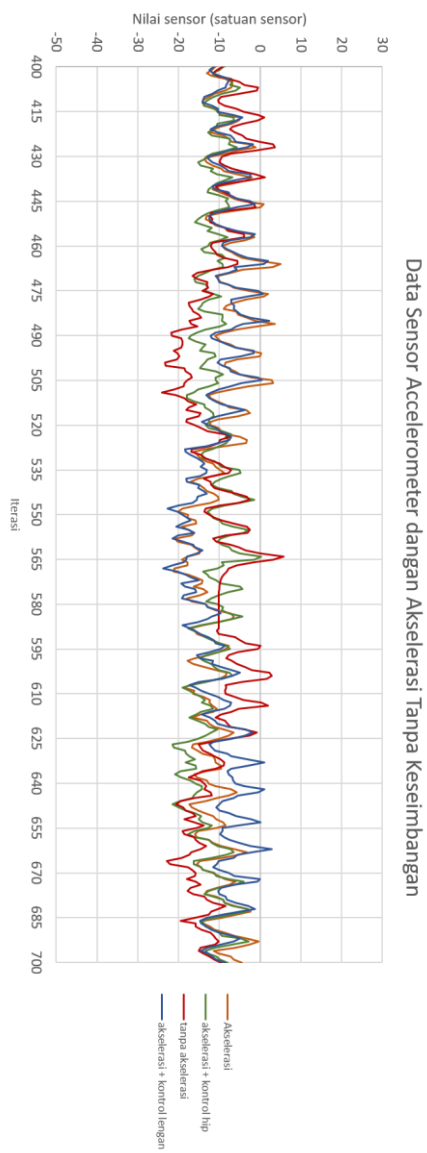
5. Perbandingan data kondisi awal berjalan robot



6. Perbandingan data robot saat deselerasi



7. Perbandingan berjalan robot saat akan berhenti



BIODATA PENULIS



Penulis lahir di Sleman, 22 Juli 1994 dan diberi nama Anas Mufid Nurrochman oleh kedua orang tuanya. Penulis memulai jenjang pendidikan di sekolah dasar di SD Negeri Suryodiningratan 3 pada tahun 2001 hingga 2007. Penulis melanjutkan pendidikan tingkat menengah di SMP Negeri 5 Yogyakarta pada tahun 2007 hingga 2010. Kemudian penulis

melanjutkan jenjang pendidikan di SMA Negeri 1 Yogyakarta pada tahun 2010 hingga 2013. Pada tahun 2013, penulis memulai pendidikan strata I di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selama kuliah, penulis aktif mengikuti kegiatan perlombaan robotika regional, nasional, dan internasional serta menjadi asisten laboratorium elektronika dasar.

Email : anasmfd@gmail.com

Halaman ini sengaja dikosongkan.